UNIVERSITÉ DE MONTRÉAL

CLASSIFICATION OF SCOLIOTIC DEFORMITIES FROM EXTERNAL SURFACE OF THE TRUNK BY USING SUPPORT VECTOR MACHINES

LIGEN WANG

DÉPARTEMENT DE GÉNIE INFORMATIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AVRIL 2003

Your file Votre référence
ISBN: 0-612-86435-9
Our file Notre référence
ISBN: 0-612-86435-9

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CLASSIFICATION OF SCOLIOTIC DEFORMITIES FROM EXTERNAL SURFACE OF THE TRUNK BY USING SUPPORT VECTOR MACHINES

présenté par : WANG Ligen

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. TORRES-MORENO Juan-Manuel, Ph.D., président

Mme CHERIET Farida, Ph.D., membre et directrice de recherche

M. GRANGER Louis, M.Sc., membre et codirecteur de recherche

M. LABELLE Hubert, M.D., membre et codirecteur de recherche

M. MEUNIER Jean, Ph.D., membre externe

To my family

ACKNOWLEDGEMENTS

My deepest gratitude goes to Farida Cheriet, my thesis advisor. She helped me tremendously to become a better researcher, and provided constant inspiration, feedback and support, as well as a wealthy of intriguing ideas. It was a great honour and a real pleasure for me to work with her. I sincerely thank Louis Granger and Hubert Labelle for acceptance to be my co-advisors. They gave me many helpful comments on the thesis. I also want to thank the members of my thesis committee, Juan-Manuel Torres-Moreno and Jean Meunier, for their patience in reading drafts of my thesis, and their valuable feedback.

Special thanks to Rui Zhang for many inciting conversations and good research ideas, for being a constant friend in good and bad time. His help with constructing surface fitting to raw scanned data points is greatly appreciated.

Thanks to Philippe Poncet for beneficial correspondences. His help with clarifying the characteristics of Calgary data is greatly appreciated.

The LIS3D research group at Sainte Justine Hospital was a stimulating and fun research environment. I especially want to thank Marie Beausejour for preparing Brace data for me and many useful suggestions and comments on my work. Thanks to Julie Joncas, Eltha Parfait, Kajsa Duke, Luc Duong and Gnahoua Zoabli for many inspiring conversations, for their constant friendship and support.

Finally, a special thank to my family, who made many sacrifices over time to help me reach this point. Their love, support and encouragement has been invaluable over the years. This thesis is dedicated to them.

RÉSUMÉ

La scoliose est une déformation tridimensionnelle de la colonne vertébrale, qui implique une rotation transversale des vertèbres pouvant entraîner une cyphose thoracique et /ou une lordose lombaire. Afin de réduire le risque de cancer lié à une dose cumulative de rayons X infligée au patient pour la détection de cette anomalie et le suivi de sa progression, des méthodes non invasives basées sur l'observation de l'asymétrie de la surface externe du tronc ont été explorées. Cependant, il a été difficile de déterminer la nature de la relation complexe et non–linéaire entre les déformations intrinsèques de la colonne vertébrale et celles propagées à la surface externe du tronc à travers les tissus mous. La meilleure approche de solution à ce problème est la méthode GA-ANN proposée par Jacob Jaremko qui a utilisé l'algorithme génétique (GA) et les réseaux de neurones artificiels (ANN) pour interpréter la relation complexe entre les déformations de la surface externe et celles des structures osseuses sous-jacentes. Le GA a servi au choix d'un ensemble d'indices parmi tous ceux qui ont été extraits à partir du modèle de la surface externe du tronc. Ensuite les ANN ont été utilisés pour corréler ces indices avec la valeur de l'angle de Cobb mesuré sur les radiographies pour caractériser la déformation de la colonne vertébrale. Bien que la méthode GA-ANN s'avère expérimentalement excellente, il subsiste tout de même certaines limites, dues à la phase indispensable d'extraction d'indices et à l'utilisation des ANN. Pour pallier à ces limites une approche de solution plus générale au problème est proposée dans le cadre de ce mémoire de maîtrise. Les Machines à Vecteurs de Support (SVM) sont utilisées pour la classification des déformations de la colonne vertébrale à partir des données de topographie de surface décrivant le modèle surfacique du tronc. Les différences fondamentales entres la méthode proposée et celle de Jacob Jeremko sont, en premier lieu, la méthode proposée n'extrait aucun indice, et en deuxième lieu, une approche SVM plutôt que ANN est utilisée pour obtenir de meilleures performances de généralisation. Les données acquises sont approximées par une B-Spline définie par un

ensemble de points de contrôle. Une technique linéaire de réduction de dimension, analyse des composantes principales, est également utilisée.

La méthode proposée a été évaluée sur deux d'ensembles de données: la surface décrivant la géométrie de la matrice de pression utilisée pour mesurer l'effet d'un corset sur un patient et la surface externe du tronc de patients acquise à partir de caméras laser à Calgary. Le premier ensemble contient les données de 41 patients et le second contient 115 acquisitions de surfaces dont certaines concernent le même patient examiné à différents instants. Les patients ont été regroupés en plusieurs classes suivant la valeur de l'angle de Cobb mesuré. Sur l'ensemble des données de corset, deux classes ont été définies suivant le critère Seuil: angle de Cobb moyen de tous les patients dans l'ensemble des données. Sur l'ensemble des données de Calgary, trois classes ont été définies (angle inférieur a 30^{o} ; compris entre 30^{o} et 50^{o} ; supérieur a 50^{o}) qui correspondent aux différents cas de gravité: patients de déformation légère, moyenne et grave. Les meilleurs résultats obtenus sur l'ensemble des données de corset étaient 0% pour l'erreur d'entraînement et 29.27% pour l'erreur de test avec le noyau de ERBF. Sur l'ensemble des données de Calgary, les meilleurs résultats obtenus étaient 9.71% pour l'erreur d'entraînement et 32.97% pour l'erreur de test avec le noyau RBF. Ce résultat est donc relativement faible par rapport aux résultats de la méthode GA-ANN. Nous pensons que la cause principale provient du fait que les points de contrôle ne sont pas une représentation invariante de la géométrie externe des patients et qu'ils ne caractérisent pas la déformation externe aussi efficacement que les indices calculés à partir des données brutes. L'effet de sur-apprentissage qui apparaît dans les résultats obtenus est lié à la petite taille de l'échantillon de données et pourrait probablement être réduit si plus de données de patients sont collectées. De plus, les divers bruits existants dans les données ont sérieusement affecté la performance de la méthode proposée. Par exemple, des corsets ont été prescrits à la majorité des patients considérés, et le traitement par corset peut altérer biomécaniquement la relation normale entre la déformation de la colonne

vertébrale et celle du tronc. Enfin, une méthode de construction d'un ensemble de SVMs a été également développée et évaluée sur des données disponibles à l'université de Californie pour tester des techniques d'apprentissage. Les résultats obtenus sont prometteurs et indiquent qu'une approche basée sur un ensemble de SVMs pourrait améliorer les résultats de classification.

ABSTRACT

Scoliosis is a common 3D spinal deformity that is biomechanically coupled with a transverse rotation of the vertebral bodies and may be accompanied by abnormal kyphosis of the thoracic spine and /or lordosis of the lumbar spine. In order to reduce the cancer risk associated with the series of full-torso X-rays which are used for the detection of this disease and monitoring its progress, non-invasive methods which make use of the torso surface asymmetry caused by scoliosis have been introduced. However, the nature of the complex, non-linear relation between surface and spinal deformities has been difficult to determine. The most successful method on this subject in the past was the GA-ANN method introduced by Jacob Jaremko, in which he utilized genetic algorithms (GA) and artificial neural networks (ANN) to interpret the complex relations between surface scans and spinal deformity. GA was used to select most suitable features that were extracted from the surface model, and ANN was used to correlate these features to Cobb angle (the measurement of spinal deformity). Although GA-ANN method achieved excellent performance in experiments, the limitations existing in feature extraction and ANN turned us to look for other more general solutions to the problem. This project approached the old problem of estimating the severity of scoliotic deformity from torso surface with Support Vector Machine (SVM), a type of novel learning methodology. The main differences between our method and Jaremko's were that firstly, we did not extract any feature from torso surface; secondly, we employed SVM instead of ANN in order to have better generalization performance. The scanned data points were represented by the control points of the surface that fitted to these scanned raw points. A linear dimension reduction technique, principal component analysis, was also involved.

We tested our method on two types of datasets: Brace dataset and Calgary dataset (containing 41 and 115 data, respectively), on which we committed the classification

experiments. The patients were divided into several classes according to their Cobb angle. On Brace dataset, two classes were defined by using the mean Cobb angle of all patients in the dataset as the threshold. On Calgary dataset, three classes (Cobb angle < 30^{o}, 30-50^{o}, and > 50^{o}, respectively) were defined which corresponded to patients having mild, moderate, and severe spinal deformity, respectively. The best results obtained on Brace dataset were 0% training error and 29.27% test error with ERBF kernel. The best results obtained on Calgary dataset were 9.71% training error and 32.97% test error with RBF kernel. Comparing to the result of GA-ANN method, this result is relatively poor. We think the main cause is from the nature of control points that are not a steady and invariant representation of surface deformities, and it does not capture the deformity information as accurately as features. The '*overfitting*' effect in our results is common with small sample sizes and would likely be reduced as more patients' data are collected. The various '*noises*' existing in the datasets seriously affected the performance of our method too, for instance, most patients in the datasets were braced, while bracing mechanically altered the normal relation between torso surface and spinal deformities. When more patients' data of better quality will be collected, our method can be expected to perform better. We also developed a method of constructing an ensemble of SVMs and test it on some artificial data sets. The results were rather promising and showed potential for a future study.

CONDENSÉ

Introduction

Dans le cadre de ce projet de maîtrise une nouvelle méthode d'estimation de la déformation de la colonne vertébrale à partir de la géométrie externe de la surface du tronc de patients atteints de scoliose a été développée. La scoliose est une déformation tridimensionnelle de la colonne vertébrale, essentiellement visible grâce à une courbure latérale de la colonne et est associée à une asymétrie du tronc et de la cage thoracique. Les patients sont actuellement suivis en mesurant la progression de la courbure de la colonne vertébrale (elle est mesurée à l'aide l'angle de Cobb) à partir d'une série de radiographies. Plusieurs chercheurs ont révélé qu'une dose cumulative de rayons X peut augmenter de manière significative les risques de plusieurs types de cancer pour les enfants atteints de cette maladie. Donc, il est indispensable de réduire ce risque en clinique, par exemple en faisant appel à une évaluation non invasive de la progression de la courbe à partir des changements visibles sur la surface externe du tronc. La déformation du torse est généralement le premier indice qui pousse à diagnostiquer une scoliose et reste le signe le plus important de scoliose pour les patients. Cependant la relation entre le tronc et la déformation de la colonne vertébrale est complexe et difficile à décrire analytiquement. La méthode GA-ANN (Genetic Algorithm - Artificial Neural Network), développée par Jacob Jaremko à l'Université de Calgary, est l'approche la plus complète et récente, dans laquelle la sévérité de la scoliose, mesurée par l'angle de Cobb, est estimée grâce à un réseau de neurone artificiel. Un algorithme génétique est utilisé pour sélectionner le meilleur sous-ensemble d'indices à présenter à l'entrée du réseau à partir d'un ensemble d'indices décrivant l'asymétrie de la surface du torse. Le réseau de neurone catégorise 83 parmi 89 données d'entraînement (93%) et 24 parmi 26 données testées (92%) comme ayant des courbes légères, modérées ou sévères (angle de Cobb $<30^{\circ}$, $30\text{-}50^{\circ}$, $>50^{\circ}$ respectivement). Leurs résultats suggèrent que la mesure de

l'asymétrie du torse est suffisante pour évaluer la sévérité de la maladie des patients atteints de scoliose à l'aide d'un réseau de neurone artificiel.

Bien que la méthode GA-ANN est très prometteuse et a amélioré les tentatives précédentes, explorées par d'autres chercheurs, pour établir une relation entre le tronc et la déformation de la colonne vertébrale, elle possède encore des limites. Les principales sont reliées à l'utilisation d'un réseau de neurone et l'extraction d'indices pour caractériser la déformation de la surface externe du tronc. L'entraînement efficace d'un ANN requière des indices qui décrivent la déformation de la surface aussi précisément que possible. Même après avoir intégré des milliers de points acquis du tronc dans plusieurs douzaines d'indices d'asymétrie, nous sommes encore face au défi de la sélection des indices les plus appropriés comme entrée au ANN. D'autres limitations incluent, par exemple, l'entraînement du réseau de neurone et l'algorithme génétique sont très coûteux; le temps nécessaire pour la sélection de l'index dans l'algorithme génétique est inacceptable lors d'une utilisation clinique de cette méthode ; le réseau de neurone peut seulement trouver un minimum local, mais pas le minimum global. Pour pallier aux limites citées une nouvelle approche a été explorée dans le cadre de ce projet de maîtrise.

Objectif

L'objectif de ce mémoire est de développer un schéma général, visant à éviter les limitations de la méthode GA-ANN, pour estimer les déformations scoliotiques à partir de la surface externe du tronc sans une procédure préalable d'extraction d'indices. Nous remplaçons ANN par SVM (Support Vector Machine) pour établir un lien entre la géométrie 3D de la surface externe du tronc et la déformation de la colonne vertébrale. L'objectif à long terme de notre étude est de réduire l'utilisation des rayons X et d'augmenter la fréquence du suivi de la progression de la scoliose en clinique.

Méthode

Au lieu d'extraire des indices à partir des données acquises, nous proposons des techniques d'approximation de surfaces afin de réduire l'ensemble des données acquises. Les points de contrôle de la surface d'approximation peuvent décrire la surface d'intérêt, et par conséquent remplacer les nombreux points acquis. Avec cette technique, la dimension des données peut être réduite de centaines de milliers à quelques centaines (cela dépend du niveau de précision avec lequel nous souhaitons représenter la surface). Nous pouvons aussi réduire davantage la dimension de l'ensemble des points de contrôle en utilisant une technique d'analyse par composantes principales (PCA: Principal Component Analysis). Avec la technique PCA, la dimension des données peut être réduite à quelques douzaines et même moins (cela dépend de la variabilité que nous souhaitons conserver). Après la réduction de l'espace des données le résultat obtenu est soumis à une machine à vecteurs de support (SVM) pour la classification des données. Dans ce qui suit, nous décrirons de façon détaillée chaque étape de notre méthode.

L'approximation de la surface

Nous avons utilisé l'algorithme d'approximation globale aux moindres carrés. Nous avons modélisé une surface NURBS (Non Uniform Rational B-Splines) de degré (p,q) pour approximer les données acquises. Ayant un nombre fixe de points de contrôle (n), nous avons estimé la courbe (ou la surface) approximative décrivant les données. La procédure d'approximation est formulée comme un problème d'optimisation non-linéaire ayant comme variables les points de contrôle, les noeuds, ou les poids, afin de minimiser un type d'erreur (par exemple, les moindre carrés ou le maximum). Dans notre cas, les seules variables sont les points de contrôles dont le nombre total est fixé, et la technique des moindres carrés est utilisée pour résoudre le problème d'optimisation obtenu. La courbe non-rationnelle de degré p vérifiant la relation ci-dessous:

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \quad u \in [0,1]$$

est déterminée en considérant:

- $Q_0 = C(0)$ et $Q_m = C(1)$;
- et Q_k s'expriment au sens des moindres carrées, comme la solution qui réalise le minimum de

$$\sum_{k=1}^{m-1} \left| Q_k - C(\bar{u}_k) \right|^2$$

par rapport aux (n+1) variables, P_i; les $\{\bar{u}_k\}$ sont les valeurs des paramètres calculé préalablement. Alors :

$$R_k = Q_k - N_{0,p}(\bar{u}_k)Q_0 - N_{n,p}(\bar{u}_k)Q_m \quad k = 1, \cdots, m-1$$

Et

$$f = \sum_{k=1}^{m-1} \left| Q_k - C(\bar{u}_k) \right|^2 = \sum_{k=1}^{m-1} \left| R_k - \sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k) P_i \right|^2$$

$$= \sum_{k=1}^{m-1} \left[R_k \cdot R_k - 2\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k)(R_k \cdot P_i) + \left(\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k) P_i \right) \cdot \left(\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k) P_i \right) \right]$$

f est une fonction des variables $P_1, \cdots, P_{n-1}$. Nous appliquons la technique standard linéaire des moindres carrées afin de minimiser la fonction f. Les dérivés de f par rapport aux $n-1$ variables, P_l, sont supposées nulles. La l^e dérivée est

$$\frac{\partial f}{\partial P_l} = \sum_{k=1}^{m-1} \left(-2N_{l,p}(\bar{u}_k) R_k + 2N_{l,p}(\bar{u}_k) \sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k) P_i \right)$$

ce qui implique que

$$-\sum_{k=1}^{m-1} \left(N_{l,p}(\bar{u}_k) R_k + \sum_{i=1}^{n-1} \sum_{i=1}^{n-1} N_{l,p}(\bar{u}_k) N_{i,p}(\bar{u}_k) P_i \right) = 0$$

Par conséquent

$$\sum_{i=1}^{n-1} \left(\sum_{i=1}^{n-1} N_{l,p}(\bar{u}_k) N_{i,p}(\bar{u}_k) \right) P_i = \sum_{k=1}^{m-1} N_{l,p}(\bar{u}_k) R_k$$

$l = 1, \cdots, n-1$ forment le système de $n-1$ équations avec $n-1$ inconnues

$$(N^T N) P = R$$

Où N est la matrice de dimensions $(m-1) \times (n-1)$

$$N = \begin{bmatrix} N_{1,p}(\overline{u}_1) & \cdots & N_{n-1,p}(\overline{u}_1) \\ \vdots & \ddots & \vdots \\ N_{1,p}(\overline{u}_{m-1}) & \cdots & N_{n-1,p}(\overline{u}_{m-1}) \end{bmatrix}$$

R est le vecteur de $n-1$ points

$$R = \begin{bmatrix} N_{1,p}(\overline{u}_1)R_1 + & \cdots & + N_{1,p}(\overline{u}_{m-1})R_{m-1} \\ & & \vdots \\ N_{n-1,p}(\overline{u}_1)R_1 & \cdots & + N_{n-1,p}(\overline{u}_{m-1})R_{m-1} \end{bmatrix} \text{ et } P = \begin{bmatrix} P_1 \\ \vdots \\ P_{n-1} \end{bmatrix}$$

Les points de contrôle à déterminer sont la solution de l'équation décrite ci-dessus.

Analyse par composantes principales

Le but de l'analyse par composantes principales est de déterminer une base orthogonale de vecteurs (des vecteurs propres) pour décrire un nouvel espace de représentation des données. La première composante représente la variabilité maximale des données. La deuxième orthogonale à la première représente la variabilité de seconde importance, et ainsi de suite. Dès que quelques composantes sont jugées suffisantes pour représenter la variabilité de l'ensemble des données, on peut ignorer les autres. Cela permet d'effectuer une réduction des dimensions de l'espace de représentation. Les composantes sont les vecteurs propres de la matrice de covariance de l'ensemble des données. Maintenant, on va décrire brièvement comment calculer les composantes principales. Supposons qu'on a une population x aléatoire, où

$$x = (x_1, \ldots, x_n)^T$$

et la moyenne de cette population est

$$\mu_x = E\{x\}$$

et la matrice de covariance de ces données est

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\}$$

Les composantes de C_x, dénotées par c_{ij}, représentent les covariances entre les variables aléatoires x_i et x_j. La composante c_{ii} est la variance de la variable x_i. Si les variables x_i

et x_j ne se corrèlent pas, leur covariance est zéro ($c_{ij} = c_{ji} = 0$). La matrice de covariance est toujours symétrique. Étant donnée une matrice symétrique comme la matrice de covariance, on peut calculer une base orthogonale puis trouver ses vecteurs propres et les valeurs propres associées. Les vecteurs propres e_i et les valeurs propres correspondantes λ_i sont les solutions de l'équation

$$C_x e_i = \lambda_i e_i, \ i = 1,...,n$$

Si on met les vecteurs propres en ordre décroissant, on pourra former une base orthogonale. Le premier vecteur propre aura la direction de la plus grande variance de données. Donc, on peut trouver les directions selon lesquelles l'ensemble des données a la valeur d'énergie la plus significative.

Étant donné un ensemble de données dont on a déjà calculé la moyenne de l'échantillon et la matrice de covariance. Soit A est une matrice composée des vecteurs propres de la matrice de covariance. La transformation d'un vecteur x dans le nouvel espace de représentation est décrite comme suit :

$$y = A(x - \mu_x)$$

Les composants de y sont les coordonnées dans la base orthogonale. Au lieu d'utiliser tous les vecteurs propres de la matrice de covariance, on peut représenter les données sous forme de quelques vecteurs de la base orthogonale. Si la matrice A_K est restreinte aux K premiers vecteurs propres, la transformation est formulée comme suit :

$$y = A_K (x - \mu_x)$$

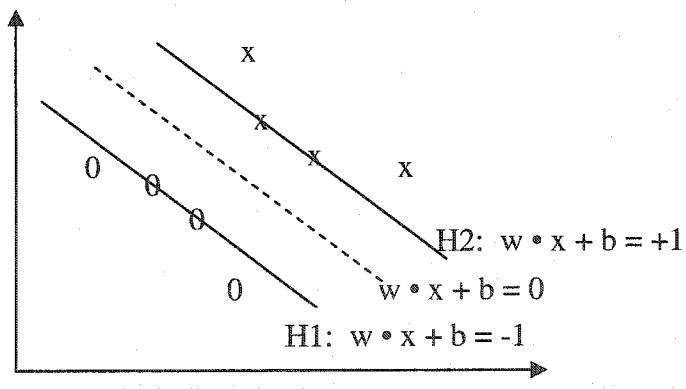
Alors, on projette le vecteur original sur le système de coordonnées de dimension K. Cela minimise l'erreur par moindre carrée entre les données originales et leur représentation dans le nouvel espace.

Si les données se concentrent dans un espace linéaire, une réduction de l'espace des données sera effectuée sans trop de perte d'information. En choisissant un nombre fixe

des vecteurs propres, et leurs valeurs propres associées, on peut obtenir une représentation consistante, ou une abstraction des données. Dans le cadre de ce projet, un taux de variabilité de 80% et 90% a été retenu. Nous avons aussi effectué les expériences sans PCA afin d'évaluer l'impact de cet étape sur le processus de classification.

Machines à vecteurs de support

Dans cette section, nous décrivons l'algorithme de classification binaire non séparable à l'aide d'un SVM linéaire. Le cas séparable est seulement un cas particulier. Étant donnés un ensemble de points $x_i \in R^n$ avec i = 1, 2,..., N. Chaque point appartient à l'une ou l'autre des deux classes et ceci est exprimé par $y_i \in \{-1, +1\}$. Le but est d'établir l'équation d'un hyperplan qui divise l'ensemble des points de façon à laisser tous les points de la même classe du même côté tout en maximisant la distance entre les deux classes et l'hyperplan. L'équation de l'hyperplan de séparation peut être écrite de la façon suivante $w \cdot x + b = 0$ (voir la figure suivante):



H1 et H2 sont des hyperplans parallèles à l'hyperplan de séparation. L'hyperplan de séparation se situe au milieu de H1 et de H2. Ainsi les équations décrivant les plans H1 et H2 sont respectivement : $w \cdot x + b = -1$ et $w \cdot x + b = +1$. Nous tenons compte du bruit, ou de la séparation imparfaite. C'est-à-dire, nous n'imposons pas strictement des points de repères entre H_1 et H_2, mais nous voulons plutôt pénaliser les points de repères qui

croisent les extrémités. La pénalité C sera finie (si $C = \infty$, on revient au cas séparable). Nous considérons des variables de relaxation non négatives $\zeta_i \geq 0$ pour tenir compte du bruit. Le but des variables ζ_i est de tenir compte d'un nombre restreint de points mal classifiés. Si les points de repères sont linéairement séparables, alors ζ_i est nul. Puis les équations des plans H1 et H2 deviennent:

$$w \cdot x_i + b \geq +1 - \zeta_i \quad for\ y_i = +1,$$
$$w \cdot x_i + b \leq -1 + \zeta_i \quad for\ y_i = -1,$$
$$\zeta_i \geq 0, \quad \forall i.$$

et nous ajoutons à la fonction objective une limite de pénalité:

$$\underset{w,b,\zeta}{\text{minimize}} \frac{1}{2} w^T w + C(\sum_i \zeta_i)^m$$

où *m* est habituellement placé à 1, ce qui nous donne

$$\begin{aligned} \underset{w,b,\zeta_i}{\text{minimize}} \quad & \frac{1}{2} w^T w + C(\textstyle\sum_{i=1}^N \zeta_i) \\ \text{sujet } de \quad & y_i (w^T x_i - b) + \zeta_i - 1 \geq 0, \quad 1 \leq i \leq N \\ & \zeta_i \geq 0, \qquad\qquad 1 \leq i \leq N \end{aligned}$$

En utilisant les multiplicateurs de Lagrange α, β, le lagrangien est:

$$\begin{aligned} \ell(w,b,\zeta_i;\alpha,\beta) &= \frac{1}{2} w^T w + C\sum_{i=1}^N \zeta_i \\ &\quad - \sum_{i=1}^N \alpha_i [y_i (w^T x_i - b) + \zeta_i - 1] - \sum_{i=1}^N \mu_i \zeta_i \\ &= \frac{1}{2} w^T w + \sum_{i=1}^N (C - \alpha_i - \mu_i)\zeta_i \\ &\quad - \left(\sum_{i=1}^N \alpha_i y_i x_i^T \right) w - \left(\sum_{i=1}^N \alpha_i y_i \right) b + \sum_{i=1}^N \alpha_i \end{aligned}$$

Ni les ζ_i's, ni leurs multiplicateurs de Lagrange n'apparaissent dans le problème dual de Wolfe:

$$\underset{\alpha}{\text{maximize}}\, \ell_D \equiv \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

sujet à:

$$0 \le \alpha_i \le C,$$

$$\sum_i \alpha_i y_i = 0.$$

La seule différence avec le cas parfaitement séparable est que α_i est maintenant borné par C au lieu de ∞. La solution est de nouveau donnée par

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

La plupart des α_i sont nuls, donc le vecteur w est une combinaison linéaire d'un faible pourcentage de l'ensemble des points originaux. Ces points se nomment des vecteurs de support parce qu'ils sont les points les plus proches de l'hyperplan de séparation et les seuls points requis pour déterminer l'hyperplan de séparation. Etant donné un vecteur de support x_j, le paramètre b peut être obtenu à partir des conditions de KKT tel que

$$b = y_j - w \cdot x_j$$

Le problème de classifier un nouveau point est maintenant simplement résolu en considérant

$$w \cdot x + b$$

Par conséquent, les vecteurs de support contiennent toute l'information nécessaire pour classifier de nouveaux points.

Résultats

Nous avons effectué des expériences pour deux ensembles de données: les données de surface de la matrice de pression acquises à l'Hôpital Saint Justine et celles de surface externe du tronc acquises à l'université de Calgary. Nous avons effectué des expériences en appliquant une technique PCA et sans appliquer une technique PCA sur les données

soumises au SVM. Dans les cas ou PCA a été appliquée, nous avons retenu un taux de 80% et 90% de variabilité des données. Nous avons effectué sur les données de la matrice de pression des expériences d'entraînement et de test, ainsi que celles de la validation croisée et la stratégie 'Leave one out'. Nous avons constaté que la stratégie 'Leave one out' est meilleure que la validation croisée pour les petits groupes de données. Par conséquent, la stratégie 'Leave one out' a été retenue pour les données de surface du tronc de Calgary. En ce qui concerne les données de la matrice de pression, les résultats sont décrits ci-dessous:

Kernel	Erreur d'entraînement	Erreur de test	Vecteurs de support
Linéaire	3.12%	65.85%	58.72%
Polynomiale	0%	39.02%	92.5%
RBF	39.39%	78.05%	80%
ERBF	**0%**	**29.27%**	100%

Table: Les résultats de classification avec 90% PCA et la stratégie 'Leave one out'.

Kernel	Erreur d'entraînement	Erreur de test	Vecteurs de support
Linéaire	19.88%	53.66%	91.95%
Polynomiale	0%	34.15%	98.11%
RBF	37.99%	75.61%	94.15%
ERBF	**0%**	**29.27%**	100%

Table: Les résultas de classification sans PCA mais avec la stratégie 'Leave one out'.

En ce qui concerne les données acquises à Calgary, les résultats obtenus sont décrits ci-dessous :

Kernel	Erreur d'entraînement	Erreur de test	Vecteurs de support
Polynomial 16x8	6.29%	34.07%	73.52%
RBF 16x8	**9.71%**	**32.97%**	73.89%
Polynomial 41x11	3.2%	47.25%	76.32%
RBF 41x11	5.69%	45.05%	76.57%
Polynomial 61x31	0%	42.86%	73.38%
RBF 61x31	0%	41.76%	72.82%

Table: Les résultats de classification sans PCA avec la stratégie 'Leave one out'

Discussion

Les résultats obtenus ne sont pas aussi bons que ceux de Jaremko pour les données de Calgary. Il y a plusieurs raisons pour cela. Le problème principal vient du fait que les points de contrôle sont utilisés comme représentation de la surface. Les résultats des expériences nous montrent que les points de contrôle n'ont pas pu caractériser la surface aussi bien que les indices calculés par Jaremko. Nous nous sommes aussi rendus compte que coordonnées 3D des points de contrôle ne sont pas invariantes par rapport aux différentes tailles de patients. Les dimensions de l'ensemble de données étaient relativement grandes. Bien que nous avons utilisé une technique de réduction des dimensions, le PCA, nous n'avons pas réussi à résoudre le problème. Noua avons aussi perdu des informations pendant la procédure de la réduction des dimensions.

Nous avons proposé une nouvelle méthode pour résoudre le problème d'estimation de la sévérité de la déformation de la colonne vertébrale à partir de données de la surface externe du tronc. La méthodologie proposée n'effectue aucune extraction d'indices, mais utilise les points de contrôle du modèle surfacique du tronc comme représentation. Un SVM est utilisé pour la classification des déformations scoliotiques.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

MFMER	Mayo Foundation for Medical Education and Research
NSF	National Scoliosis Foundation
PA	Posterior-Anterior
SRS	Scoliosis Research Society
ANN	Artificial Neural Network
GA	Genetic Algorithm
SVM	Support Vector Machine
PCA	Principal Component Analysis
ML	Machine Learning
3D	Three Dimension

INTRODUCTION

Scoliosis is a lateral curvature, greater than 10 degrees, of the spine. The lateral deviation is biomechanically coupled with a transverse rotation of the vertebral bodies and may be accompanied by abnormal kyphosis of the thoracic spine and /or lordosis of the lumbar spine (Haasbeek 1997). In clinics, scoliosis is primarily measured by the Cobb angle. The propagation of scoliosis on the external surface of patient's trunk is the asymmetry of the trunk and rib cage. Most curves can be treated nonoperatively if they are detected before they become too severe. However, 60% of curvatures in rapidly growing prepubertal children will progress and can cause pain, osteoarthritis, disability or even respiratory collapse if untreated. Therefore, scoliosis screening and monitoring are necessary and patients should be examined every 6 – 9 months. Currently, the progression of scoliosis is mostly monitored by physical exam and radiography. Radiation exposure in girls with adolescent idiopathic scoliosis has been reported to increase their risk of reproductive pathology. Multiple X-rays can significantly increase the risk of several types of cancer for these children with scoliosis (Levy et al., 1996). Although new low-dose digital X-ray devices have been shown to significantly reduce radiation exposure (Kalifa et al., 1998), new non-radiographic, non-invasive techniques that can be used in conjunction with radiographs may serve to further decrease associated radiation risks and hence have very important significance for the clinic.

The deformation appearing on the torso surface of the scoliotic patient gives people a heuristic that the torso asymmetry may be used to assess the severity of the scoliosis. In fact this is a pattern recognition problem. Numerous methods, such as Moire photography, modeling and imaging tools, knowledge discovery from scoliosis database, local centroids evaluation, Quantec analysis which uses contour mapping methods etc., have been developed to quantify scoliosis from this approach (Denton 1992; Sakka et al. 1997). However, the precise relation between spinal and surface deformity is unknown.

This relation is very complex and difficult to describe analytically because the spinal deformity is translated into surface asymmetry via the rib cage, with its incompletely understood mechanical properties, as well as spinal muscles, viscera, fat, and skin (Closkey et al., 1993; Stokes et al., 1989; White et al., 1990). Therefore, this complex spine-surface relation is extremely difficult to model directly (e.g., through a finite element model). So, it will not be surprising that all these methods have their own strength and weakness, and none of them have gained domination in popularity after taking into account all sorts of factors, such as performance, cost, complexity etc.

In very recent years, a type of novel and innovative methodology has been introduced into the field of scoliosis research. Dr. Jacob Jaremko and his team from the University of Calgary used learning methodology to approach the solution of this problem, i.e., estimating scoliosis severity from surface asymmetry. Classical programming techniques cannot solve this problem, since no mathematical model of the problem is available. Under this kind of situation, the learning methodology is of strategic importance and could provide the key to its solution. In the method used by Jacob Jaremko, an artificial neural network, a type of learning system, was used to relate torso asymmetry to spinal deformity. They acquired X-rays and 115 360-degree torso surface scans from 48 scoliosis patients to develop a genetic algorithm-artificial neural network (abbreviated as GA-ANN from now on). That network would recognize and predict the Cobb angle of scoliotic spinal deformity in patients. They used various indices such as age, curve direction, and bracing status as inputs for the neural network. The neural network using the indices selected by genetic algorithm estimated the Cobb angle within 5° in 65% of the test set and 84% of the training set, and within 10° in 85% and 99% respectively. The neural network also categorized 83 out of 89 training-set records (93%) and 24 out of 26 test-set records (92%) as having mild, moderate or severe curves (Cobb angles $<30^{\circ}$, $30\text{-}50^{\circ}$, $>50^{\circ}$ respectively). Their results suggest that determination of torso asymmetry alone, through use of an artificial neural network, appears to be effective to assess

severity of disease in patients with scoliosis. This technique may also help to reduce the need for the spinal x-rays often necessary in these patients.

Although the GA-ANN method has been very successful and outperformed previous attempts to relate surface and spinal deformity, there still exist some limitations in it. The main limitations are related to the '*natural born*' constraints existing in the mechanism of the neural network and the use of features (i.e., indices) to represent the surface deformity. Effective ANN training requires a set of input indices that describes the surface deformity as completely and as efficiently as possible. Even after integrating thousand of raw torso surface data points into several dozen asymmetry indices, we are still faced with the challenge of selecting the most appropriate of these indices to use as ANN inputs. Some other limitations include, for instance, both neural network training and genetic algorithm are very time-consuming; the extra processing time required for genetic-algorithm index selection would be undesirable in clinical use of this method; neural network can only find local minima, not global minima; and etc. So, although the GA-ANN method was very successful, we still would like to try other approaches to the scoliosis estimation problem.

Hence, based on these facts, another kind of method whose starting point was to avoid the above-mentioned constrains of GA-ANN method was developed, for the goal of estimating scoliosis severity from trunk surface deformity. Firstly, we did not want to do feature extraction, i.e., we wanted to make full use of the raw torso geometrical data points directly. Secondly, we wanted to replace neural network by other sort of learning machine which does not have those 'natural-born' constraints existing in the mechanism of neural network. With these two considerations, we came to our project – estimation of scoliosis severity from the torso surface by support vector machine. Support vector machine (SVM) is a type of novel and extremely powerful learning machine. It meets many of the challenges confronting machine learning systems. The four problems of

efficiency of training, efficiency of testing, overfitting and algorithm parameter tuning are all avoided in the design of SVM.

This thesis is organized into five chapters. Introduction section gives a general introduction to the investigated problem, the difficulties, our general objective and the methodology we employed to solve it. Chapter 1 introduces the basic background knowledge of scoliosis. Literature investigation of previous attempts on our problem and the comparison of those methods and the proposed method are also included. Chapter 2 aims at describing the techniques involved in our project, including principal component analysis, and support vector machine. Chapter 3 deals with experiments we committed and results we obtained. In chapter 4 we present an investigative study which is our preliminary try on constructing the ensemble of SVMs. Conclusion is the final section presenting the overall conclusions and suggestions for future development of the described work.

CHAPTER 1 - STATE OF THE ART

1.1 Scoliosis

1.1.1 Definition

On an X-ray taken from behind, a normal spine appears straight. However, a spine affected by scoliosis looks like an "S" or a "C", showing a lateral deviation of the normal vertical line of the spine (Figure 1.1). This condition of side-to-side spinal deformity is called scoliosis. Some of the bones in a scoliotic spine also may have rotated slightly, making the person's waist or shoulders appear unbalanced.

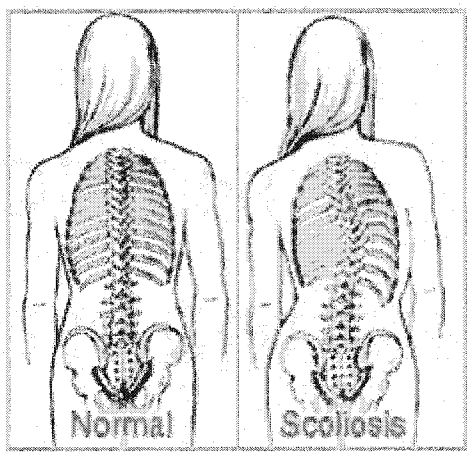


Figure 1.1: Posterior-anterior (PA) view of normal and scoliotic spine (*SRS Patient Handbook, 2002*)

Scoliosis is a kind of widely existing disease. According to past research, one in 10 persons will have Scoliosis. Two to three persons in every 1000 will need active treatment for a progressive condition. In one out of every 1000 cases, surgery may be necessary (NSF, 2002). Due to the prevalence of this disease, many efforts have been spent on the research from different points of view.

1.1.2 Spinal Anatomy

For the convenience of narration, we introduce the terminologies used in the spinal anatomy at first. A human spine is composed of 24 vertebrae, plus the sacrum and tailbone (Figure 1.2). These 24 vertebrae can be divided into cervical, thoracic, and lumbar sections.

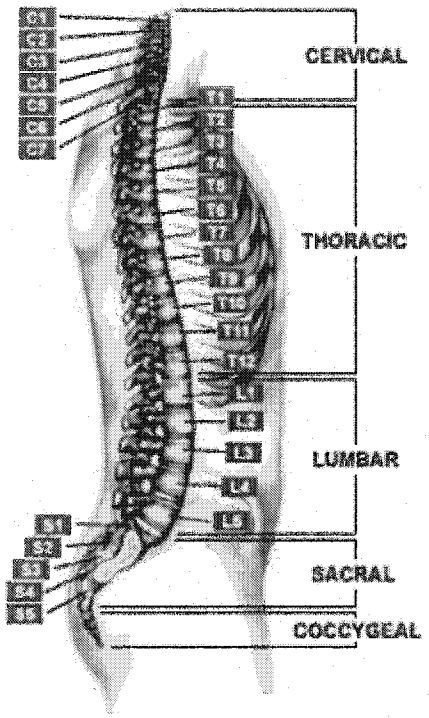


Figure 1.2: Spinal anatomy (taken from www.espine.com)

These characteristics of spine can be concluded in the following table:

Term	# of Vertebrae	Body Area	Abbreviation
Cervical	7	Neck	C1 – C7
Thoracic	12	Chest	T1 – T12
Lumbar	5 or 6	Low Back	L1 – L5
Sacrum	5 (fused)	Pelvis	S1 – S5
Coccyx	3	Tailbone	None

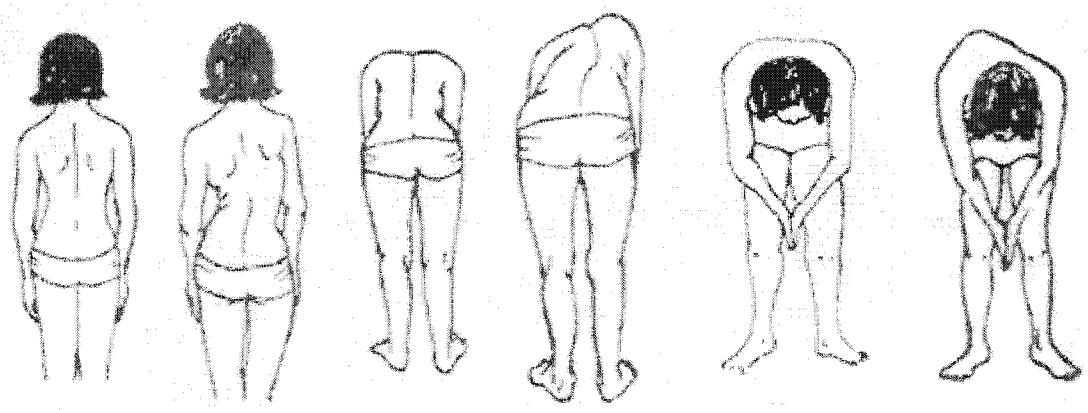
Table 1.1 Characteristics of spine (Keith Bridwell, 2001)

1.1.3 Signs and Symptoms

The following are the most common symptoms of scoliosis. However, each individual may experience symptoms differently. Symptoms may include:

- Unbalanced shoulders
- Prominent shoulder blade or shoulder blades
- Unbalanced waist
- Elevated hips
- Leaning to one side

These symptoms can be visually percepted at clinics, as shown in the following figures:



(a) (b) (c) (d) (e) (f)

Figure 1.3: Comparisons of normal and scoliotic patients (NSF resources)

(a) Normal (b) Scoliotic (c) Normal (d) Scoliotic (e) Normal (f) Scoliotic

1.1.4 Types of Scoliosis

The types of scoliosis can be divided into three main categories (Julie Joncas, 2000).

- **Idiopathic scoliosis**

In more than 80% of the cases, a specific cause is not found and such cases are termed idiopathic, i.e., of undetermined cause. This is particularly so among the type of scoliosis seen in adolescent girls.

- **Congenital scoliosis**

This type of scoliosis exists from birth and is visible on an X-ray film. It is secondary to a vertebral deformity.

- **Other types of scoliosis**

Comparing to the above two types, other types of scoliosis are relatively rare to be observed at clinic, so we can include all of them as one type, such as neuromuscular scoliosis, traumatic scoliosis, and iatrogenic scoliosis.

1.1.5 Causes

As we have mentioned above, in most (80 to 85 percent) cases, the cause of scoliosis is unknown - a condition called idiopathic scoliosis. This is particularly so among the type of scoliosis seen in adolescent girls. Conditions known to cause spinal deformity are congenital spinal column abnormalities, neurological disorders, genetic conditions and a multitude of other causes. Scoliosis does not come from carrying heavy things, athletic involvement, sleeping/standing postures, or minor lower limb length inequality (SRS).

1.1.6 Type of Curve Patterns

According to the location of the curvature occurred on the spinal curve line, four common types of curve patterns seen in scoliosis may be defined:

- Thoracic – 90% of the curves occur on the right side.

Apex of curve is between T2 and T11.

- Lumbar – 70% of the curves occur on the left side.

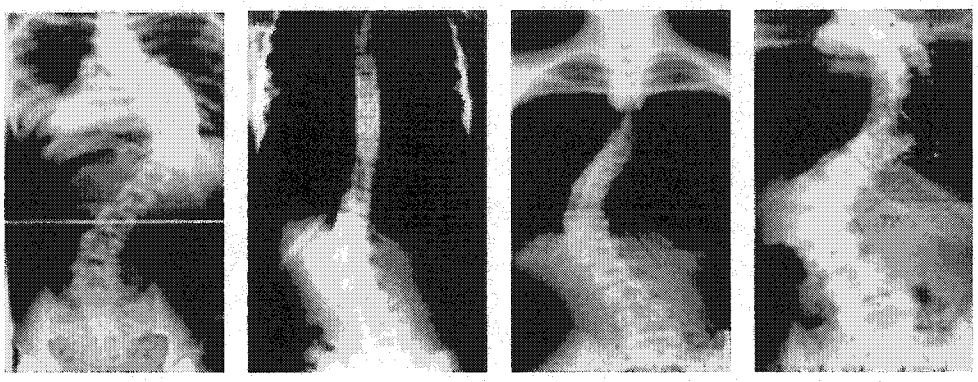
Apex of curve is at L2 or L3.

- Thoracolumbar – 80% of the curves occur on the right side

Apex of curve is at T12 or L1.

- Double major – 90% of the curves have a right thoracic convexity along with a left lumbar convexity.

There are two large structural curves in both the thoracic and lumbar spine.



(a) (b) (c) (d)

Figure 1.4: Four types of curve patterns (taken from www.espine.com)
(a) Thoracic (b) Lumbar (c) Thoracolumbar (d) Double major

1.1.7 Scoliosis Classification

There are a number of criteria existing in scoliosis classification. King's criterion is the most commonly accepted one in clinics. King developed a classification scheme for idiopathic thoracic and combined thoracolumbar scoliosis (King et al., 1983). It was based on the curves described by Moe (Moe, 1958) and is used to help guide management including placement of Harrington rods. The definition of each category of scoliosis and its features are summarized in the following table:

Curve	Vertebra	Other Features	*Type*
S-shaped with both thoracic and lumbar curves	both curves cross midline with lumbar curve larger than the thoracic curve on standing X-rays	flexibility index < 0	I
	both curves cross midline with lumbar curve smaller than or equal to the thoracic curve on standing X-rays	flexibility index >= 0	II
	thoracic curve crosses midline but lumbar curve does not ("overhanging")		III
single long thoraco-lumbar curve	L5 centered over sacrum; L4 tilts into curve		IV
S-shaped primarily involving thoracic curve	T1 tilts into the convexity of upper curve; L2-L5 centered over sacrum	upper curve structural on bending	V

Table 1.2: King's classification

Flexibility index:

• The flexibility of the thoracic and lumbar curves is measured on maximum lateral bending.

• Subtracting the correction of the thoracic curve from the correction of the lumbar curve is termed the flexibility index.

The following figure illustrates the five categories of scoliosis according to King's classification scheme.

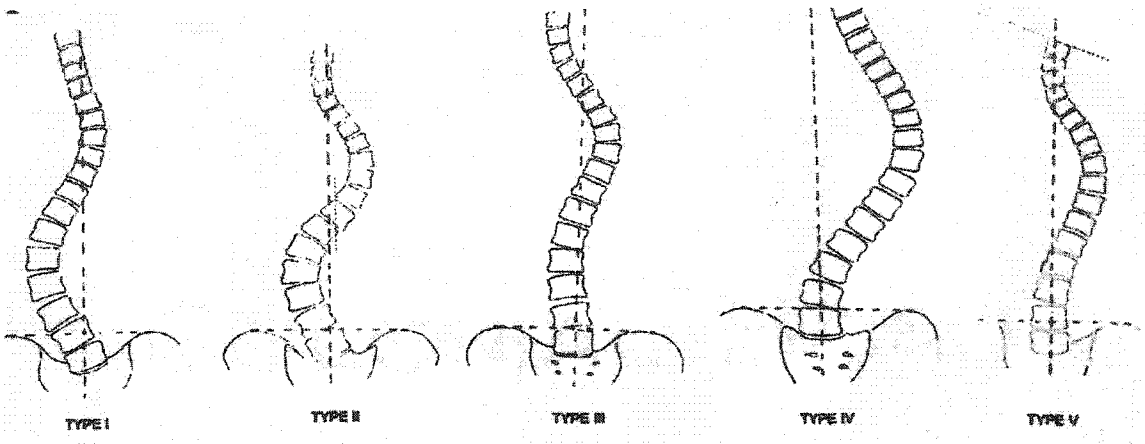


Figure 1.5: King's classification (King et al., 1983)

1.1.8 Curve Progression

Although most scoliosis is of unknown cause, there does appear to be a developmental connection in many cases. Most cases of scoliosis occur just before and during adolescence, when children are going through a growth spurt. Risk factors for curve progression include (Stuart L. Weinstein):

Curve pattern

In general, double curve patterns are at greater risk for progression than single curve patterns. Lumbar curves tend to have the least risk of progression of all curve patterns.

Age

The younger the child when scoliosis appears, the greater the chance of curve progression.

Menarche

A curve detected prior to menarche has a much greater chance of progression (66%) than one detected post menarche (33%).

Curve magnitude

The larger the initial curvature at detection, the greater the chance of progression.

Sex

Girls are 10 times more likely to experience curve progression than boys.

1.1.9 Clinical Treatment

Treatment choice in adolescent idiopathic scoliosis is determined by a complex equation which includes the patient's physiologic (not chronologic) maturity, curve magnitude and location and potential for progression. The goal of treatment is to stop the progression of the curve and prevent deformity. Treatment may include:

- **Observation and repeated examinations**

Small curves measuring less than 20-25 degrees that do not require brace treatment should be observed during periodic examinations of four to six months or 1 year intervals based on their size. Observation remains a form of treatment because any 5 degree increase in the size of the curve may change the course of treatment.

- **Bracing**

Bracing may be used when the curve measures between 25 to 40 degrees on an x-ray, and during skeletal growth. The type of brace and the amount of time spent in the brace will depend on the severity of the condition.

- **Surgery**

Doctors typically recommend surgical treatment for patients whose curves are greater than 40 to 50 degrees.

- **Other approaches**

Some cliniciens have tried electrical stimulation of muscles, chiropractic manipulation and exercise as ways to treat scoliosis. There's no evidence that any of these methods will prevent spinal curvature from progressing (SRS).

1.2 Analysis of Internal Spinal Deformity

For the purpose of analysis, a set of indices which are used to quantify the deformity of the spine have been developed (Jaremko, 2001).

1.2.1 Cobb Angle and Its Variants

Cobb angle is the most commonly used index in the quantification of scoliosis. It is measured from a 2D X-ray. Cobb angle measures the angle of curvature of the spine. To use the Cobb method, one must first determine which vertebrae are the end-vertebrae of the curve. These end-vertebrae are the vertebrae at the upper and lower limits of the curve that tilt most severely toward the concavity of the curve. Once these vertebrae have been selected, the angle between intersecting lines drawn perpendicular to the upper endplate of the superior vertebrae and the lower endplate of the inferior vertebrae is the Cobb angle (see figure 1.6).

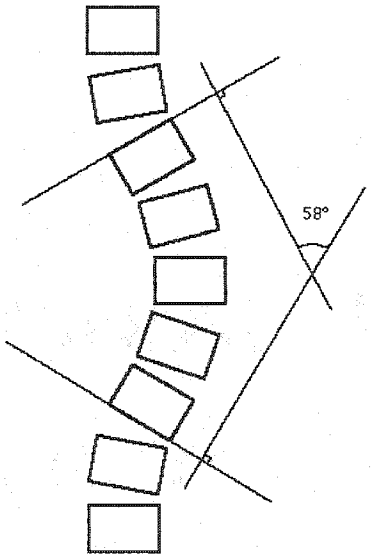


Figure 1.6: Cobb method for measurement of scoliosis (Michael L. Richardson, 2000)

In the above figure, the Cobb angle of the spine is 58 degree.

An important problem with the Cobb angle is that it is measured on the frontal or lateral plane projection of a spine. That is to say, it reflects only the 2D shape of the spine. As we have known, scoliosis is a 3D deformation problem. It is primarily a lateral deviation asymmetry, but accompanied with axial rotation produced by some inherent mechanism in intervertebral joints (motion segments) of the spine (Stokes et al.). So it is not

surprising that the Cobb angle cannot sufficiently and accurately reflect the spinal deformity. While the Cobb angle is the current gold standard of scoliosis monitoring, it must be remembered that it is not fundamental to the curve but is only an incomplete and imprecise measure derived from the true 3D shape of the spine.

Another alternative to the Cobb angle is the "computed Cobb angle". A smooth mathematical curve is first fitted through the frontal plane projections of the coordinates of the centers of the vertebral bodies. Inflection points of this curve are located, and the angle subtended by perpendiculars to the curve at these inflection points is measured as the "computed Cobb angle" (stokes et al. 1987). An advantage of the computed Cobb angle is its low variability measurement which is 1.2°, compared to up to 9° for the manual Cobb angle (Labelle et al., 1995a).

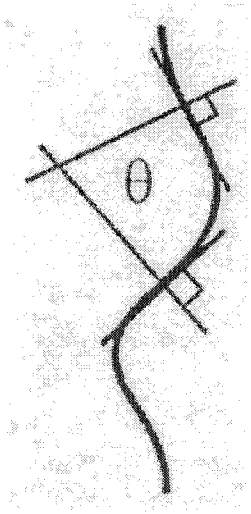


Figure 1.7: Computer Cobb angle (Jaremko, 2001)

1.2.2 Apex Location

The apex of a scoliotic curve is defined as the vertebrae with maximal lateral deviation.

1.2.3 Vertebral Rotation

One may estimate the degree of rotation of the vertebra at the apex of the curve by looking at the relation of the pedicles to midline.

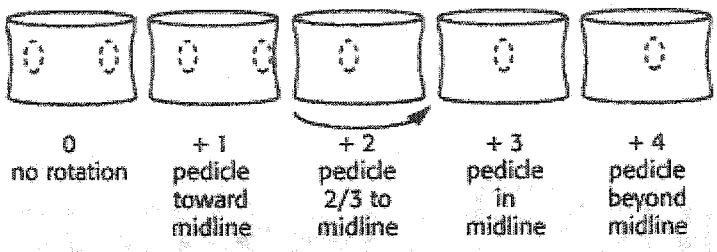


Figure 1.8: Measurement of rotational component of scoliosis (Michael L. Richardson, 2001)

1.2.4 Posterior Rib Rotation

Posterior rib rotation is measured from 3D reconstruction of the shape of the ribs, which are calculated from the digitized stereo radiograph images of the midlines of the ribs by the method of Dansereau and Stokes (Dansereau and Stokes, 1989). This method uses an iterative computer program to find a line whose projection onto the two radiographic planes best matches the digitized midlines. Posterior rib rotation is measured at each anatomical level as the axial rotation relative to the frontal plane of a line drawn tangentially to the projection of each pair of ribs onto a horizontal plane. This index of rib asymmetry is analogous to back surface rotation.

1.2.5 Rotation of Plane of Maximum Curvature

The "rotation of plane of maximum curvature" of the part of each spine defined by the end vertebrae of the scoliosis curve can be found by projecting the spine sequentially onto planes rotated about a vertical axis. A computer program calculates the spinal curvature, based on the perpendiculars to the curve at the end vertebrae, at each rotation. The axial rotation at which this curvature becomes a maximum is noted, with the

patient's sagittal plane as the origin for measurement. This index is used to describe the rotation of the spine.

1.3 Analysis of External Trunk Deformity

For the purpose of analysis, a set of indices which are used to quantify the deformity of the trunk has been developed (Jaremko, 2001).

1.3.1 Back Surface Rotation

Adam's forward bending test is used for detecting rotational asymmetries on the back surface of scoliotic patients by clinicians. It requires no additional equipment (such as a scoliometer or humpometer) and can help to identify scoliosis. The test is accomplished by having the patient bend forward at the waist standing with feet together and the knees straight. The patient's arms are dependent and the hands are held with the palms opposed. The examiner looks along the horizontal plane of the spine from the back and side to detect an asymmetry in the contour of the back. A rotational deformity known as a "rib hump" (arrow) can be easily identified.

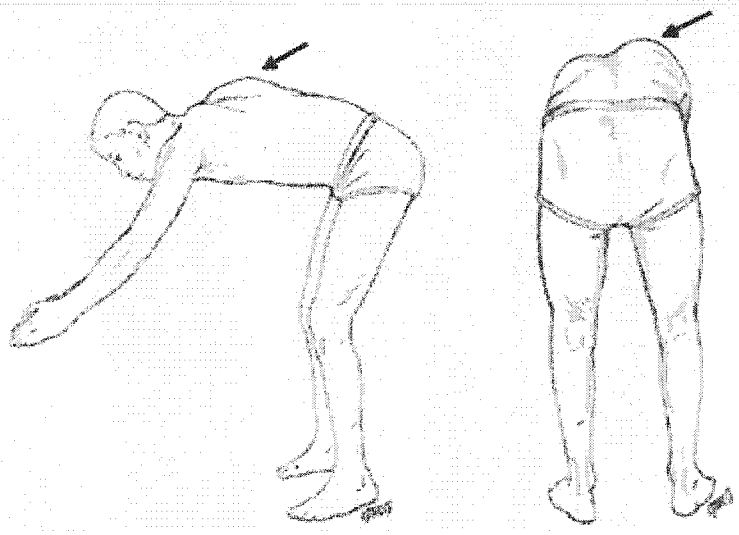


Figure 1.9: Adam's forward bending test (Gilbert M. Gardner, 2001).

Back surface rotation describes the rotation of a horizontal plane cross section made through the back surface at the level of each vertebra. It is defined as the angle of the line between thoracic rib humps or tangent to the lumbar midline. As illustrated in the following figure (Figure 1.10), by projecting a segmental level which consists of a vertebra, the ribs and the back surface onto a horizontal plane, the back surface rotation is measured about a vertical axis, with positive value assigned to clockwise rotations as seen from above (Stokes, 1989).

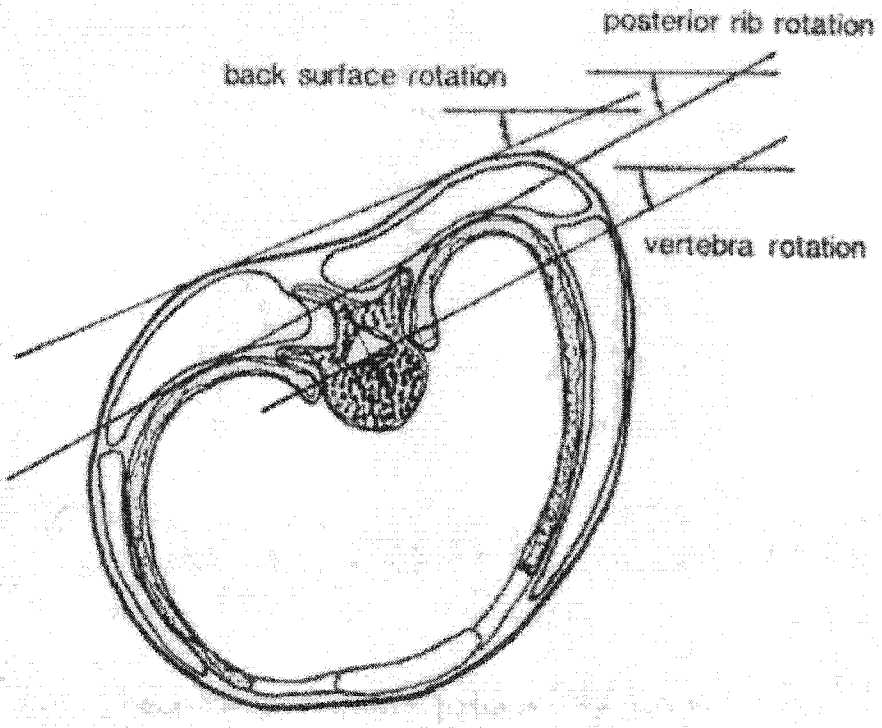


Figure 1.10: Back surface rotation, rib rotation, and vertebra rotation (Stokes, 1989)

The maximum vertebra rotation, back surface rotation, and posterior rib rotation all occur close to the apex of the scoliosis. The vertebra having the greatest axial rotation is at the apex of the scoliosis curve, or within two vertebral levels of it, as is also the maximum posterior rib rotation. The maximum back surface rotation is found to be at a level between two vertebrae above the apex and three vertebrae below it (Stokes, 1989).

1.3.2 Apex Level

Surface asymmetry is greatest near the apex of spinal deformity in scoliosis, generally speaking within two vertebral levels of the radiographic curve apex (Scutt et al., 1996; Stokes and Moreland, 1989). This maximal lateral deviation of the trunk is termed as the apex level of the surface. It can be used in surface topography to estimate spinal apex location relatively accurately.

1.3.3 Spinous Process Line

The line of spinous processes, palpable just below the surface of the back and usually visible in slender patients, is the most direct available evidence of the position of the underlying spine. The "spinous process angle" can be calculated from this line by the same method as the computer-Cobb angle (section 1.2.1). This angle is smaller than the Cobb angle, likely because forces in the posterior spinal ligaments and muscles straighten the spine from behind by encouraging vertebrae to rotate into the concavity of a scoliotic curve. Still, the spinous process angle correlated closely in most cases to the Cobb angle with $r = 0.77 - 0.94$ (Drerup and Hierholzer, 1996; Herzenberg et al., 1990; Letts et al., 1988; Turner-Smith et al., 1988; Weisz et al., 1988; Wong et al., 1997). The spinous process angle has been described as the best single index of scoliosis available from surface topography (Turner-Smith et al., 1988).

1.3.4 Trunk Indices

While most study of surface deformity in scoliosis has focused on the shape of the back, the recent availability of 360° full-torso surface scan data has enabled study of torso cross-sectional asymmetry. Many indices might be extracted from these horizontal slices of the trunk. For example, one group defined the angle formed analogously to the Cobb angle from the line of trunk cross-section centroids as the "torsographic angle", and they found that it correlated well to the Cobb angle ($r = 0.69 - 0.87$) in 93 patients (Dawson et al., 1993). Other indices, such as trunk centroid offset, intrinsic cross-sectional rotation, shear, and size asymmetry can also be defined as features used in surface topography.

1.4 Scoliosis Assessment

There are primarily two types of approaches in assessing the severity of scoliosis. One approach is to check the spine directly. X-ray is the typical representative of this type of method. The other approach is to try to estimate the interior spinal deformity by analyzing the external torso surface deformity which is caused by the interior deformity. Both approaches have their own advantages and disadvantages.

1.4.1 X-ray Evaluation

Today, in clinics scoliosis is mostly evaluated by measuring the lateral curvature of the spine on an X-ray that is taken with the patient facing the X-ray film (the anterior to posterior, or AP view – as opposed to a lateral view). When a curve is present, it is measured and discussed in terms of Cobb angle. The severity of the spinal deformity of patient with scoliosis is determined by the Cobb angle. The main advantage of X-ray method is its accuracy of measuring the severity of the scoliotic deformity since the shape of the spine can be presented directly. A possible upper limit to the accuracy of Cobb angle estimation is its wide measurement variability of up to 9^{o} between different observers and up to 5^{o} with the same observer (Goldberg et al., 1988). The disadvantages of X-ray are also evident. The first limitation of X-ray assessment to the scoliosis is from the Cobb angle index employed by the X-ray method. There are some issues existing today about the completeness and representativeness of this index. The main problem with the Cobb angle is that it is only a two dimensional measurement, nevertheless scoliosis is a three dimensional deformity. But basing on the reality that no other better indices have been invented so far, Cobb angle is still the most widely accepted and the most important index used in clinics. So, in our work we still adopted the Cobb angle as the measurement of the severity of the scoliotic deformity. There are still some other limitations of X-rays. For instance, the X-rays are costly and risky. High exposure to ionizing radiation particularly in adolescents is not acceptable since there is strong evidence of increased carcinogenic risk (Nash et al., 1979). In a study of women with

scoliosis seen between 1935 and 1965, the incidence of breast cancer was nearly double that of the general population (Hoffman et al., 1989). Although radiation exposure to the breasts, thyroid and gonads has been reduced by modern X-rays that use lower doses of radiation and are taken from behind the patient (posterior-anterior, PA) rather than from in front (anterior-posterior, AP), the lings and bone marrow receive more radiation by PA X-ray than by AP X-ray (Levy et al., 1996; Pope et al., 1984). Scoliosis patients typically receive an average of 12 full spinal X-rays during their adolescent growth years, increasing the incidence of cancer of the breast, thyroid, lung, ovary and bone marrow by up to 2.4 cases per 1000 AIS patients (Levy et al., 1996). The limitations of radiographic evaluation of scoliosis, and the high cost and current limited availability of magnetic resonance imaging (MRI), have encouraged research into surface assessment of scoliotic deformity.

1.4.2 Surface Topography Evaluation

Surface Topography is rapidly becoming an essential component in the comprehensive assessment of 3D spine deformities. It has long been accepted that conditions such as scoliosis are 3D in nature and that traditional methods of measuring spinal curvature such as Cobb angle using X-rays do not give a 3-D measure of the back surface. The use of surface topography for the assessment of scoliotic deformity in the clinic depends firstly on the quality of measures which reliably characterize deformity of the back, and secondly on the easiness and speed with which these measures can be applied.

Many non-invasive methods have been developed to measure and assess back shape and posture. These can be divided into two categories: tactile and non-tactile. Tactile methods are low in cost and simple to use but are limited in the measurements that can be made and generally can only document curves either in the sagittal or transverse section of the back. The contour tracer (Thulborne et al., 1976), Bunnell scoliometer (Bunnell, 1984), spinal pantograph (Willner, 1981) and flexirule (Lovell et al., 1989) are the most common tactile methods in widespread use. Non-tactile methods are more costly because

they are generally based on optical techniques. Although accurate they tend to be cumbersome to move and expensive to maintain. Moiré (Moreland et al., 1983; Tartaro et al., 1986), ISIS (Oxford metrics, 1987; Turner-Smith, 1988) and Quantec (Wojcik et al., 1994) are the most widely used however, owing to their cost and complexities are usually only accessible to specialist units. Many clinicians who would benefit from such a system currently still have to rely on qualitative visual methods to assess posture and shape.

Methods of quantifying scoliosis deformity from back or trunk surface asymmetry have been introduced since the 1970's. Moiré Fringe Contouring was the first one of them and was widely applied in various applications. Moiré fringes are formed when one line or grid pattern is superimposed upon a similar line or grid pattern, as shown in the figure below:

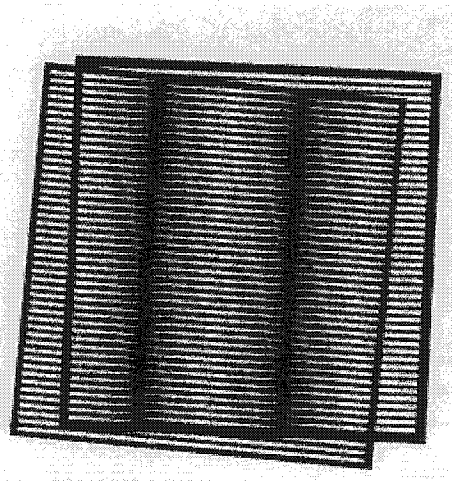


Figure 1.11: Superimposed gratings (taken from www.scoliosis-world.com)

A grating is projected onto the object and an image of the object formed in the plane of a reference grating. The interaction of the superimposed projection grating lines with the reference grating causes moiré fringes to be produced which appear superimposed on the surface of the object being measured. As the projected grating is distorted by the

irregularities in the shape of the object's surface, the resulting fringe pattern describes surface contours.

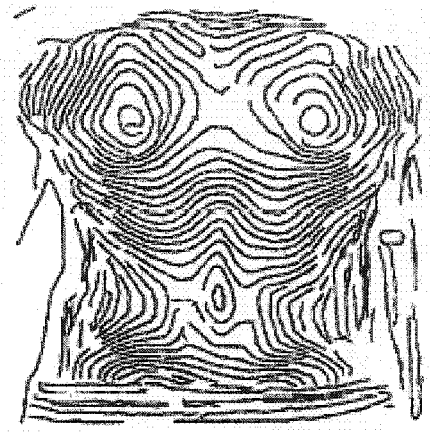


Figure 1.12: Moiré fringes (taken from www.scoliosis-world.com)

The use of moiré fringes to acquire 3D surface shape information was well established. Their application to the measurement of areas of the human body began with the work of Hiroshi Takasaki as early as 1973 when he successfully applied moiré topography to the measurement of the human body for medical purposes. In 1994, two researchers from Algeria built a computer vision system for diagnosing scoliosis (Batouche M and Benlamri R., 1994), in which moiré images of the patient's back are extracted using the infinite size symmetric exponential filter first, then image interpretation, i.e., scoliosis diagnosing, is performed using some relevant features and the three dimensional surfaces that are reconstructed from the contour fringes by applying a parallel relaxation operator. In order to have a real-time vision system, most of the system's components are implemented in a parallel fashion. The experimental results have shown that this system is robust to noise and is reliable for the recognition of most scoliosis deformities. Many works have been carried out in scoliosis research by using moiré fringes. Asymmetric Moiré topography is a sensitive marker of scoliosis, but the false-positive rate can go as

high as 505 (Stoke and Moreland, 1989; Willner and Willner, 1982), the patterns produced changed easily with small changes in patient position (Moreland et al., 1981).

Besides Moiré contour topography, another category of surface topographic method is the back surface raster scan. The principal idea of this method is to record the intersection lines of a horizontal beam of projected light and the back surface first, then compute cross-sectional back surface coordinates. The scan results can be manipulated by computer to generate back surface asymmetry indices for the purpose of evaluating internal spinal deformity. This process was once extremely time-consuming by manual measurement (Thulbourne and Gillespie, 1976), but modern raster photogrammetric methods are rapid and accurate (Vandegriend et al., 1995) (Figure 1.13).

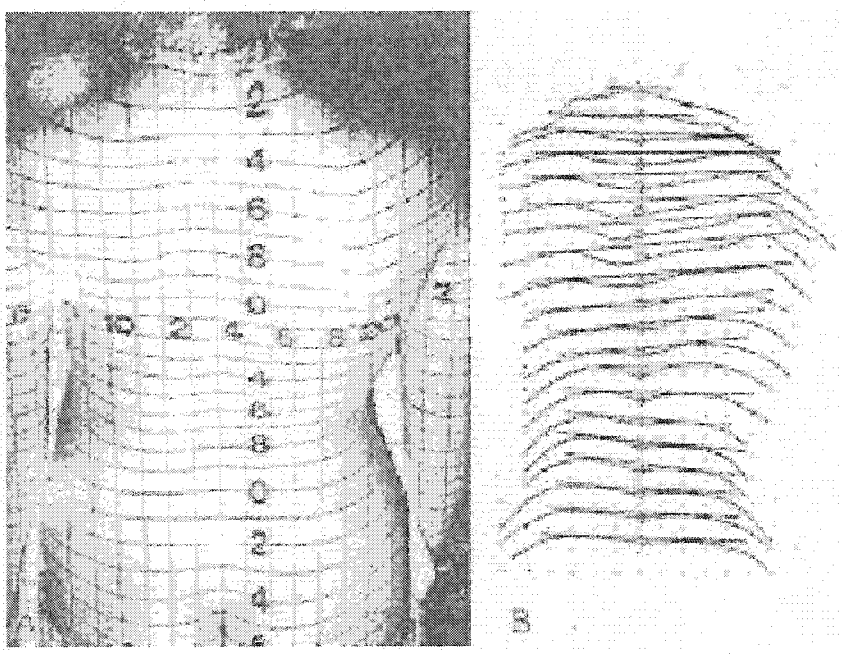


Figure 1.13: Back surface raster scan (Stokes and Moreland, 1989)

A widely used example system of this approach was ISIS, the Integrated Surface Imaging System (Theologis et al., 1997; Tredwell and Bannon, 1988; Turner-Smith et al., 1988; 17 Upadhyay et al., 1988; Weisz et al., 1988). It was used for screening programs and for

research into the nature of scoliosis. The scanning of the entire back can be done in <2 s (Turner-Smith et al., 1988). A recent commercial raster system is the Quantec scanner, acquiring back surface 3D coordinates in "voxel" form via a single charged-coupled-device (CCD) camera (Goldberg et al., 2001a; Liu et al., 2001; Thometz et al., 2000). The latest one is perhaps the SYDESCO, a new 3D laser-video scanner developed for trunk surface topography (Treuillet et al., 2002). It associates a fast video camera with a laser stripe light and uses the principle of triangulation-based range sensing, which is commonly used in vision machine to infer 3D shape. The trunk is scanned by vertically moving both camera and laser on a linear rail controlled by computer. Then an asymmetry index is calculated from the 3D data to detect the scoliosis. This is only a preliminary study for 3D scoliosis evaluation.

The third category of surface topographic evaluation method is the full-torso scan approach. Previous works mostly focused on the back of the surface since most deformities appear on the back. However, the value of quantifying not just the back surface but the 360° deformity of the entire scoliotic trunk is beginning to be recognized. Such a trunk scan would be most useful if it covered the entire trunk with a high resolution of data points. Not many researches have been done in this approach. The earliest one was a group in Japan using a single rotating scanner to capture the entire torso shape, though scan range was limited to 10 cross-sections in 27 cm (Dawson et al., 1993; Ishida et al., 1987; Ishida et al., 1982). Another device rotated the subject 360° on a turntable for a complete scan, but the system was slow (30 s per scan) and results of tests and correlation studies were not published (Gomes et al., 1995). A group in France made 360° torso scans using four raster cameras, but their research was intended predominantly for brace design instead of internal spinal deformity (Sciandra et al., 1995). The most recent and complete one is the scan system developed by Jaremko's group at the University of Calgary. They used four laser scanners mounted on a ring to acquire the entire torso of each patient (Poncet et al., 2000a). After the full-torso scanning was done, a comprehensive study of the features of the external shape and the correlation with the

internal spinal deformity was carried out. With the measured information from the scanned surface model, they successfully estimated the scoliotic deformity. More details about their scanning process and estimation method will be presented in later sections.

The introduction of the optical surface topography method opened new methods of description and registration of the spine status, provided possibility of examination of a large number of children during a short time. Each new technology – X-ray, Moiré-fringe and raster scans, handheld digitizers, 360°-torso scans – enables a different approach to understanding scoliosis via a different set of data describing the spine and trunk. They may be applied both for mass screening of patients with scoliosis for early detection of spinal deformity, and for following treatment monitoring.

1.5 Machine Learning in Scoliosis Research

The construction of machines with the capability of automatically learning from experience is of strategic importance, as there are many tasks which cannot be solved by classical programming techniques, since no mathematical model of the problem can be built. For example, it is not known how to write a computer program to perform handwriting character recognition, though there are plenty of examples available. It is therefore natural to ask if a computer could be *trained* to recognize the letter 'A' from examples – after all this is the way humans learn to read. This approach to problem solving is referred as the learning methodology. Machine learning (ML) provides methods, techniques, and tools that can help solving diagnostic and prognostic problems in a variety of medical domains. ML is being used for the analysis of the importance of clinical parameters and their combinations for prognosis, e.g. prediction of disease progression, extraction of medical knowledge for outcome research, therapy planning and support, and for the overall patient management. ML is also being used for data analysis, such as detection of regularities in the data by appropriately dealing with imperfect data, interpretation of continuous data used in the Intensive Care Unit, and intelligent alarming

resulting in effective and efficient monitoring. The successful implementation of ML methods can help the integration of computer-based systems in the healthcare environment providing opportunities to facilitate and enhance the work of medical experts and ultimately to improve the efficiency and quality of medical care.

Scoliosis research is certainly a sub-field of medical research. But so far, scoliosis research is not a much explored discipline yet by ML researchers. Not many ML methods have been applied into scoliosis research as in other medical applications. In this section, we summarize the main ML methods adopted in the scoliosis research so far.

1.5.1 Knowledge Discovery

Some researchers tried to solve the scoliosis classification problem with the approach of data mining (Man et al., 2000). Their method was to discover knowledge from scoliosis database using evolutionary algorithms. Two different representation of knowledge, namely rules and causal structures, were learned. Rules capture interesting patterns and regularities in the database. Causal structures represented by Bayesian networks capture the causality relationships among the attributes. Evolutional algorithms, including generic genetic programming, genetic algorithms, evolutionary programming, and evolutionary strategy, were employed to conduct the learning tasks. From the scoliosis database, they discovered knowledge about the classification of scoliosis.

According to their research, the largest effect on the clinicians from the data mining analysis of the scoliosis database was the fact that many rules set out in the clinical practice were not clearly defined. The usual clinical interpretation depends on subjective experience. This data mining effort revealed quite a number of mismatches in the classification on the type of King's curves. Their results demonstrated that the knowledge discovery process can find interesting knowledge about the data, which can provide novel clinical knowledge as well as suggest refinements of the existing knowledge.

1.5.2 Linear Discriminant Function

In 2001, a group of researchers reported a novel technique for automating human scoliosis detection by computer based on moiré topographic images of human backs (Hyoung et al., 2001). In their method, displacement of local centroids is evaluated statistically between the left-hand side regions and the right-hand side regions of the moiré images with respect to the extracted middle line. The local centroid displacement is calculated in several regions and the mean and the standard deviation of the displacement values are chosen as two features. A linear discriminant function (LDF) is defined on the two dimensional feature space based on the Mahalanobis distance and the features are classified into two categories, i.e.., normal and abnormal cases, by the LDF. The classification accuracy was 88.3% by training and testing on 120 real moiré images. Despite the relatively simple mechanism, the advantage of this technique is that it realizes simpler analysis of moiré image and, therefore, achieves much shorter processing time than any other previous method.

1.5.3 Linear Regression

A linear regression model assumes that the regression function E(Y|X) is linear in the inputs X_1, ..., X_n. In the scoliosis case, the output variable Y can be understood as the Cobb angle, the input X can be understood as weighted linear combination of torso asymmetry indices, with weights calculated to give the least squared error between the estimated and actual output. Such regression is often carried out in a stepwise logistic manner which tests all input indices one at a time, adding an index to the model if its contribution is significant and removing any indices whose contributions become non-significant (Jefferson et al., 1997). When the output is predicted not as a continuous variable but as a category (e.g., category 1 = mild curvature, category 2 = severe curvature), an equivalent technique called discriminant analysis can be employed (Liu et al., 2001). Linear regression method is simple and often provides an adequate and interpretable description of how the inputs affect the output. For prediction purpose it can

sometimes outperform fancier nonlinear models, especially in situation with small numbers of training cases, low signal-to-noise ratio or sparse data. However, given the non-linearity of many relations within the torso (e.g., ligament strains, thresholds of rotation established by bony geometry), use of linear combinations of surface indices may not be sufficient.

1.5.4 Genetic Algorithm and Neural Networks

An important contribution to scoliosis evaluation from torso surface is the work of Jacob Jaremko and his colleagues. While many investigators have focused on indices describing the asymmetry of the back surface, they found that the use of indices describing the deformity of 360° torso cross-sections improved the prediction of the Cobb angle. In their study, scoliosis severity, measured by the Cobb angle, was estimated by artificial neural networks from indices of torso asymmetry using the genetic algorithm to select the optimal set of input torso indices. The network structure they used was a 3-layer feed-forward neural network training with the standard back-propagation method. Three types of index selection techniques were applied: they were linear methods, principal component analysis, and genetic algorithm, respectively. Best performance was obtained on genetic algorithm. Basing on a data set of 115 scans of 48 scoliotic patients, they conducted two sorts of experiments: regression and classification. In the regression experiment, the neural network using the indices selected by genetic algorithm estimated the Cobb angle within 5° in 65% of the test set and 84% of the training set, and within 10° in 85% and 99% respectively. In the classification experiment, the neural network correctly classified 83 out of 89 training-set records (93%) and 24 out of 26 test-set records (92%) as having mild, moderate or severe curves (Cobb angles <30°, 30-50°, >50° respectively). Their experiments showed promise for future longitudinal studies to detect scoliosis progression without use of X-rays.

1.6 Limitations of Jaremko's Approach and another Direction

From the review of literature in scoliosis research, we have noticed that not many machine learning methods have been introduced into this field. But basing on the state of that the precise relation between spinal and surface deformities is unknown and this relation cannot be accurately mathematically modeled, machine learning methods are appropriate tools for this type of problems. In this project, we investigated the old problem of estimating scoliosis severity from trunk surface data basing on the hypothesis that changes in spinal curvature in scoliosis are related to systematic and measurable changes in trunk surface topography. Jacob Jaremko's approach is the most comprehensive and successful one so far comparing to previous works in this field. His system gives the ability for the first time of accurately predicting the severity of the spine of the patient with scoliosis, and the possibility of massively non-invasively monitoring the progression of the scoliosis. His work has significant contribution to the clinic. But some limitations still exist in his method. In order to clarify the fundamental differences between our work and his work, we discuss these limitations first.

1.6.1 Limitations of Jaremko's Approach

In his approach, he trained an ANN with a collection of input indices, which describe the torso surface asymmetry, to estimate the Cobb angle of spinal deformity for a group of scoliotic patients. There are mainly two sorts of limitations with this approach. One lies in the feature extraction, another in the neural network.

1.6.1.1 Limitations of Feature Extraction

In his method, firstly more than 250 features of torso asymmetry were collected. After optimizing the zone of calculation of each of >250 variations of torso asymmetry features and removing redundant features, 47 features of torso asymmetry (e.g., back surface rotation, apex level, and spinous process line) and clinical descriptors (e.g., age, sex, height, weight, and bracing status) remained. Then a Genetic Algorithm was utilized to

find out the '*most suitable*' set of 17 features of torso asymmetry and clinical characteristics out of the previous 47 features and clinical descriptors. Two limitations exist with this approach. Firstly it is very complicated to define and calculate these features from torso model reconstructed from the scanned spatial geometrical points; secondly even after integrating hundreds of thousands of raw torso surface data points into 17 asymmetry features, still it is not guaranteed that the remaining features represent the torso surface deformities completely and exactly. The challenge of selecting the most appropriate features to use as classifier inputs always exists there. Feature extraction also requires extremely carefully positioning of patients. A slight move of patient's posture can lead to significant change to some feature values. This is obviously disadvantageous in practice. The feature selection step by GA is extremely time-consuming in the GA-ANN method. As Jaremko mentioned in his PhD thesis: on a 650 MHz IBM PC, each neural network run took ~15 s and each generation of a genetic algorithm took ~90 min, the GA generally converged between 20 to 80 generations. That is to say, the feature selection step took 30 to 120 hours. However, we also have to point out that although some limitations exist in feature extraction approach, methods using features in pattern recognition tasks usually give better accuracy than methods which do not make use of features. This is simply because features capture information of an object more accurately than non-feature representation of the same object.

1.6.1.2 Limitations of Neural Network

Neural network is a very powerful learning machine and achieved excellent performance in Jaremko's experiments, but some limitations naturally existing in the mechanism of NN made us to consider replacing it by another type of learning machine which does not suffer from these limitations. Generally speaking, in order to acquire good generalization performance, plenty of data are usually needed for NN training and testing. But this is not our case. In reality it is very hard to collect plenty of scoliotic data for research. For instance, Jaremko's data set was from 5 data collection sessions, each of which last half a

year and it contained only 48 patients. As we know, the performance curve of a typical NN can be illustrated as in the following figure:

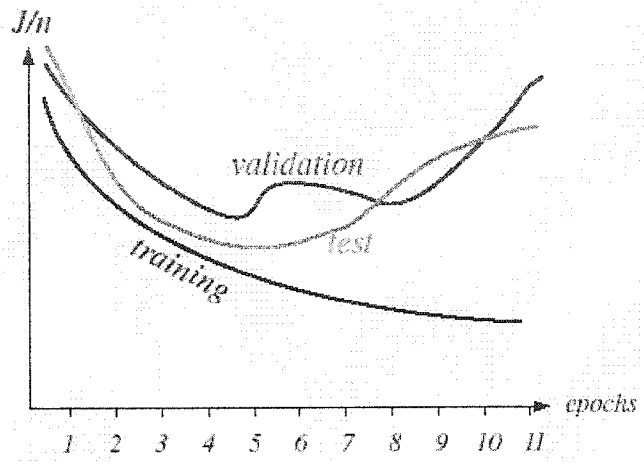


Figure 1.14: Neural Networks learning curve (Duda et al., 2001)

In order to have NN generalize well, usually a validation set has to be used in order to stop the training of the network before it becomes over adapted to the training set. The adding of a validation set makes the situation harder but unavoidable when available data are very limited. This is an undesirable characteristic of NN. Some other undesirable characteristics of NN include, for instance the fact that NN can only find out the local minimum on the error surface, it cannot find out the global minimum; each run can lead to different result; it is hard to maintain steady architecture of the NN in order to obtain optimal performance etc. These characteristics have been embedded in the design of NN. However, we also have to point out that with these limitations existing in NN does not mean at all that NN cannot perform well in practice. In fact GA-ANN method performed excellently on the Calgary scoliosis data set which is relatively small. We are merely discussing these limitations from the theoretical point of view of machine learning techniques.

1.6.2 Another Direction

Support vector machine (SVM) is a type of novel learning method. The foundation of SVM has been developed by Vapnik (Vapnik, 1995) and is gaining popularity in recent years due to many attractive features, and promising empirical performance. The formulation of SVM embodies the Structural Risk Minimization (SRM) principle, as opposed to the Empirical Risk Minimization (ERM) approach commonly employed within statistical learning methods. SRM minimizes an upper bound on the generalization error, as opposed to ERM which minimizes the error on the training data. This upper bound of the generalization error is the sum of the training error and a term which depends on the Vapnik-Chervonenkis dimension of the classifier. By minimizing the sum of both quantities, high generalization performance can be achieved. Moreover, unlike other machine learning methods, the number of free parameters in the SVM does not depend explicitly on the input dimensionality of the problem, which suggests that SVM can be especially useful in problems with a large number of inputs. Experimentally, SVM has also achieved superior performance. For example, it outperformed radial basis networks on recognizing the US postal service database of handwritten digits (Scholkopf et al., 1996) and improves the best-known result on a time series benchmark in the Santa Fe Competition by a factor of 37% (Muller et al., 1997). Besides these, it has also been successfully applied to a number of applications, ranging from time series prediction (Fernandez, 1999), to face recognition (Tefas et al., 1999), to biological data processing for medical diagnosis (Veropoulos et al., 1999). Its theoretical foundations and its experimental success encouraged us to think about the feasibility of applying SVM into our problem, i.e., estimating scoliosis severity from surface deformity. SVM has not successfully been applied into other medical applications. In this section, firstly we briefly summarize some medical applications to which SVM has been applied; then we make a comparison between SVM and ANN.

1.6.2.1 SVM in Medical Applications

The paper (Veropoulos et al., 1999) proposes an application of SVM classifiers to medical diagnosis of Tuberculosis from photomicrographs of Sputum smears. Except the fact that this is the first time that SVMs are used in a medical problem, another interesting point was the introduction of two methods that can be used for controlling the performance of the system on a particular class of the data (that is, force the SVM to better classify the data from one of the two classes of the classification task). In most medical problems, medical experts must have the ability to put more weight on one of the classes of the problem (usually the class on which the diagnosis is 'heavily' based). Another common problem in a wider area of applications is the presence of unbalanced data sets (the set of examples from one class is significantly larger than the set of examples from the other class). For these reasons, controlling the performance of a system on a particular class of the data is practically very useful. To do so, (Veropoulos et al., 1999) used a slightly modified version of the standard SVM formulation – the same idea was suggested in (Osuna et al., 1997). The idea is to use different regularization parameter C for each of the two classes. This translates in the following SVM formulation:

$$\min_{f,\xi_i} \|f\|_K^2 + C_1 \sum_{i \in class_1} \xi_i + C_2 \sum_{i \in class_2} \xi_i$$

$$\text{subject to: } y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \text{ for all } i \quad \xi_i \geq 0$$

By changing the ratio C_1/C_2, (Veropoulos et al., 1999) showed how to influence the performance of the SVM toward one of the classes, therefore altering the false negative vs false positive ratio for one of the classes. A different approach for dealing with the problem of unbalanced data or to putting more weight on one of the classes is also discussed in (Veropoulos et al., 1999).

The paper (Song et al., 2001) presents a novel landmark-based shape deformation method which provided effective solution to two problems inherent in landmark-based shape

deformation in the medical image segmentation field: (a) identification of landmark points from a given input image, and (b) regularized deformation of object shape embedded in a template. The second problem was solved using the constrained SVM regression technique, in which a thin-plate kernel was utilized to provide non-rigid shape deformations. The proposed method was applied to extract the scalp contours in cryosection head images with very encouraging results. The experiments also showed that this method was especially suitable for segmenting 3D images slice by slice, where there are only small shape variations across the neighboring slices.

In the drug design field, (Burbidge et al., 2001) reported that the SVM classification algorithm has promising potential for structure-activity relationship analysis. In a benchmark test, the SVM was compared to several machine learning techniques currently used in the field. The classification task involves predicting the inhibition of dihydrofolate reductase by pyrimidines, using data obtained from the UCI machine learning repository. Three artificial neural networks, a radial basis function network, and a C5.0 decision tree were all outperformed by the SVM. The SVM is significantly better than all of these, except a manually capacity-controlled neural network, which takes considerably longer to train.

In the paper (Chris et al., 2001), the authors studied several important issues in protein fold recognition in the context of a large number of folds, i.e., multi-class case, using support vector machines and neural networks. Most current discriminative methods for protein fold prediction use the one-against-others method, which has the well-known "false positives" problem. In this paper, the authors investigated two new methods handling with multi-class cases: the unique one-against-others and the all-against-all methods. SVM and ANN were utilized as base classifiers. Their results showed that SVM converges faster and leads to higher accuracy comparing to ANN.

1.6.2.2 SVM vs. ANN

Many classical algorithms of machine learning are able to represent any function (as for neural network, a neural network of just two hidden layers is sufficient to represent arbitrary function) and for difficult training sets will give a hypothesis that behaves like a rote learner. By a rote learner we mean one that correctly classifies the data in the training set, but makes essentially uncorrelated predictions on unseen data, i.e., *overfitting*. Generally speaking, both SVM and ANN are very powerful learning algorithms. But the formulation of SVM embodies the Structural Risk Minimization (SRM) principle, which has been shown to be superior (Gunn et al., 1997), to traditional Empirical Risk Minimization (ERM) principle, employed by conventional neural networks. Traditional neural network approaches have suffered difficulties with generalization, producing models that can overfit the data. This is a consequence of the optimisation algorithms used for parameter selection and the statistical measures used to select the '*best*' model. SVM are trained by solving a constrained quadratic optimization problem. Among others, this implies that there is a unique optimal solution for each choice of the SVM parameters. This is unlike other learning machines, such as standard Neural Networks trained using back-propagation.

ANN also suffers from the difficulty of model selection and are relatively heavy time-consuming; especially on large-scale dataset or high-dimensional dataset, the training procedure of an ANN is typically extremely slow. Comparing to ANN, SVM training is very fast. The only factor which can give limitation to SVM learning is the number of training data, this is because training of SVM leads to a quadratic optimization problem with bound constraints and one linear equality constraint. For large learning tasks with many training examples, off-the-shelf optimization techniques for general quadratic programs quickly become intractable in their memory and time requirements. But in recent years, some techniques have been invented which make large-scale SVM learning practical (Joachims 1999).

Besides the widely observed excellent accuracy of SVM, another characteristic of SVM is that it does not include *a priori* knowledge about the problem, unlike the other high performance classifiers.

At the side of experimental performance, SVM also outperforms ANN in most applications. For instance, in last section we have seen that (Burbidge et al., 2001) and (Chris et al., 2001) all reported better performance found on SVM than on ANN in drug design and protein fold recognition applications, respectively. However, there is no guarantee at all that SVM can always outperform ANN. In fact, specially designed ANN can outperform SVM. For instance, the long-term competition carried out at AT&T laboratory between ANN and SVM on NIST database shows that specially designed ANN can achieve better performance than SVM (for details see Vapnik 1998). Generally speaking the performances obtained by ANN and SVM respectively are usually very close, the difference between them is really not significant.

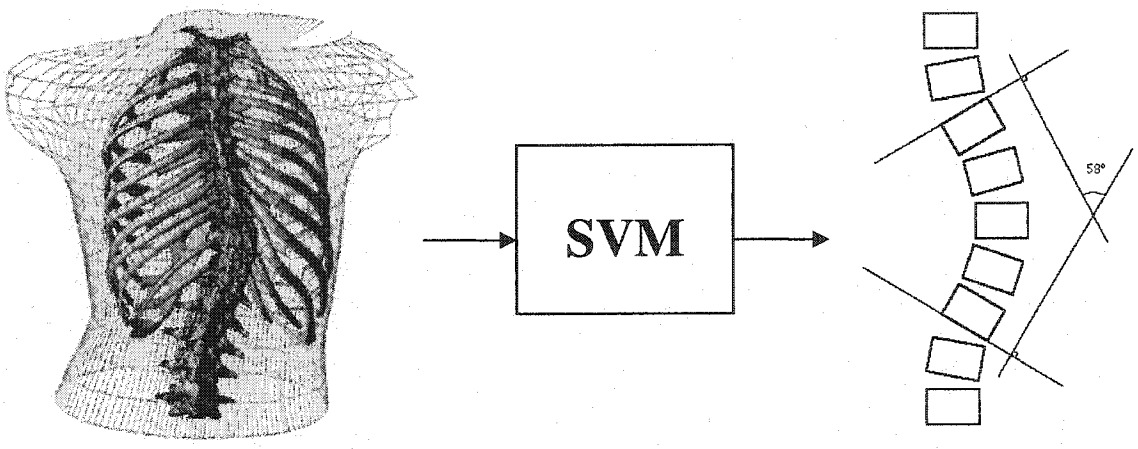
The successfulness of SVM in both theoretical and empirical sides encouraged us to apply it into our application, i.e., estimating scoliosis severity from surface deformities.

1.7 Research Question

1.7.1 Objective

We want to develop a kind of general method, which can avoid the limitations existing in the GA-ANN method used by Jacob Jaremko, for the goal of estimating scoliosis severity from trunk surface deformity. Specifically speaking, first, we do not want to do feature extraction, namely, we want to make full use of the raw torso geometrical data points directly. Second, we want to replace neural network by other learning machine which does not have those '*natural born*' constraints existing in the mechanism of neural network. We employed Support Vector Machine, a type of novel learning algorithm, to try to find out this complex non-linear relationship between spine and surface. This

project is mainly an exploratory work. The main goal is to test the feasibility and usefulness of our approach to the scoliosis estimation problem which is fundamentally different from the GA-ANN approach. The objective can be illustrated by the following figure:



Trunk Surface

(Jaremko, 2002)

Spinal Deformity
(Cobb Angle)
(Richardson, 2000)

Figure 1.15: Objective illustration

1.7.2 Long-term Goal

Scoliosis is a 3-D deformity of the spine, commonly assessed and monitored by a series of harmful X-rays in growing adolescents. The long-term goal of our method is to help to reduce the use of X-ray, and can be massively and frequently used in screening and monitoring the progression of scoliosis in clinic.

1.7.3 Difficulties

The trunk data were collected by using multiple scanners scanning patient's trunk, such that a very dense set of 3D coordinates of points of the trunk were obtained. These points uniquely represent the trunk of the patients. Theoretically, different type of deformity of

spine leads to different deformation trend on the trunk surface, hence it is possible to classify deformed internal spine from external trunk data. And theoretically, we can simply send all of these points into a classifier with unlimited computing power to do the classification. But in practice, since the data points sampled from trunk are so dense, e.g. there might be up to 65,000 points for a complete 360° full-torso scan, we have to look for data reduction processing of these points. Another difficulty is that the available amount of training data is usually very limited, as well as the dimension of each data is very high. Dimensionality reduction techniques must be involved.

CHAPTER 2 - METHODS

Instead of doing feature extraction from the scanned raw data points, we used surface fitting technique as the data reduction technique for the original dense set of points. The control points of the acquired surface can uniquely determine this surface, and consequently represent the original dense points too. With this technique, the dimension of the data can be drastically reduced from hundreds of thousands to a few hundreds (it depends on how accurate we want the surface fitting to those points). Based on the fact that the size of our data set is very small, the dimension of a few hundreds is still too high. We can continually reduce the dimension of the data set of control points by utilizing Principal Component Analysis (PCA). With PCA, the dimension of the data can be reduced to a few dozen or even less (it depends on how much variation we want to reserve). After the data set is prepared, we then send it into the SVM. The whole scheme can be illustrated by the following figure:

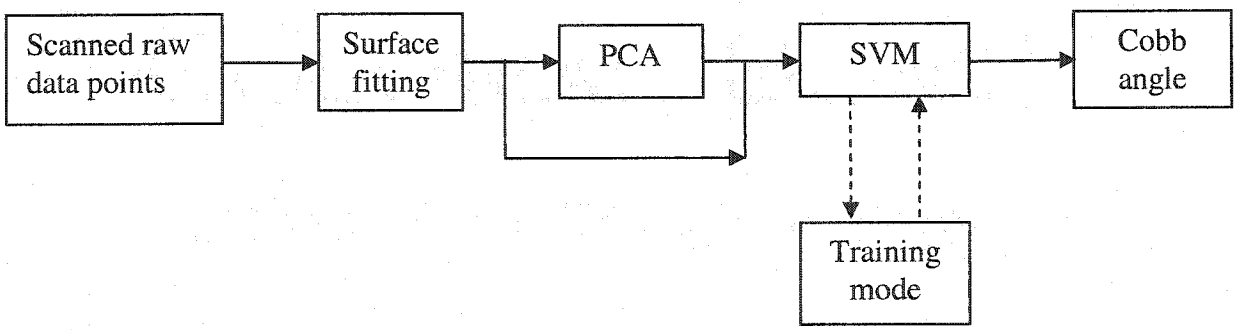


Figure 2.1: Scheme of process

In this chapter, we will describe each step of our method in detail.

2.1 Raw Data Acquisition

By comparing to the data which are finally fed into the SVM to estimate the scoliosis deformity, we address the original scanned data as the raw data, since they are acquired

directly from the surface of trunk and there was no any process imposed to them. These raw data are composed of two parts: one part is the surface data which describes the geometry of the trunk of scoliotic patients; another part is the spinal data which describes the geometry of the spine of scoliotic patients. Both data have to be recorded at the same time, i.e., when being scanned by a kind of laser scanning camera, the patient is also X-rayed at the same time. Only with this setting, each trunk shape precisely corresponds to its spinal shape. Any slight move or change in posture can cause change in the trunk shape of the patient. In order to get accurate scanning of the torso, patient's clothes should be removed, though an alternative solution, for instance a tight-fitting white top, can be available for extremely shy patients.

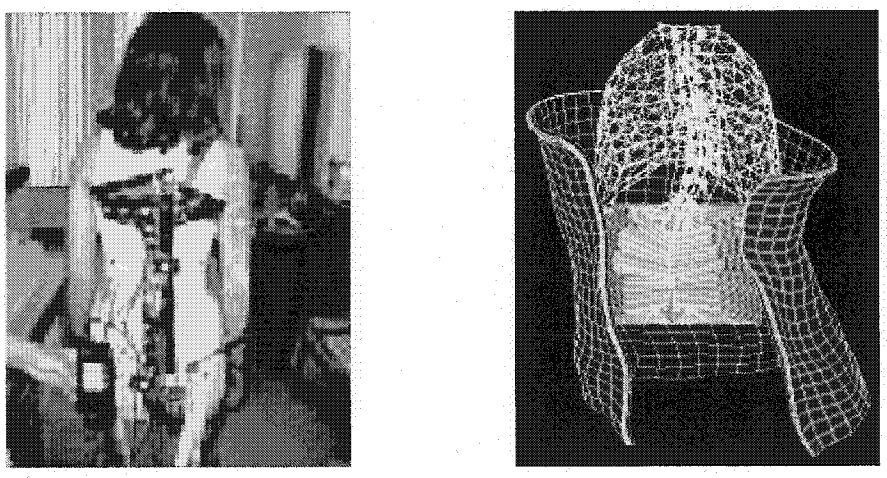
At the beginning of this project, we had no full-torso scanning data. We had only the brace data which can be seen as a kind of simulation of real full-torso 3D scanned data from Sainte Justine hospital, Montreal. Preliminary experiments were committed on the brace data. Later we got the real full-torso scanned data from Calgary University which was the same as Jaremko used in his work. So we had two sets of data in all and we committed experiments on both of them. We address them as Raw Brace data and Raw Calgary data, respectively. By contrast to the raw data, we address the processed data which were finally fed into the SVM as Brace data and Calgary data, respectively. In this part we briefly introduce how these raw data were acquired and some of their characteristics.

2.1.1 Raw Brace Data

2.1.1.1 Acquisition Technique

Patients were braced when taking X-ray. The brace was first adjusted on the patient according to the surgeon's prescription. Then a flexible latex mat was inserted at the brace-torso interface and the brace straps were tightened. 192 very thin (<2mm) polymeric pressure sensors (Verg Inc., Winnipeg, Canada) are mounted on the mat and

are used to acquire the pressure generated by the brace on the entire torso (figure 2.2). The electric wires connecting the pressure sensors were also digitized and reconstructed in 3D in order to get a 3D geometric model of the torso-brace interface for pressure area localization, and then the coordinates of the surface points were interpolated from these reconstructed wires. Thus, since the electric wires were tightly in touch with the torso surface, the points from the wires could be viewed as having the same geometrical coordinates as the surface points located right under the electric wires. These acquired points can honestly represent the shape of the trunk of the scoliotic patient.



Patient with brace Mat

Figure 2.2: Brace illustration

The acquisition of the patient's torso internal geometry was performed using a positioning apparatus and a calibration frame. Three radiographs were taken: a standard posterior-anterior (PA), a PA with a 20° angle down pitch and a lateral view. Radiographs were taken after the brace was put on. Anatomical landmarks (6 per vertebra and 11 per ribs) were digitized and reconstructed in 3D using the direct linear transformation (DLT) algorithm adapted for the spine and thorax (Dansereau et al, 1990; Marzan 1976). A geometric modeling technique that uses an atlas of already meshed

generic vertebrae was deformed to fit the reconstructed points, and used to complete the geometry (Aubin et al., 1995). These reconstructions allowed the calculation of 3-D indices characterizing their deformity (Label et al., 1996; Label et al., 1995) (computerized Cobb angles, apical axial rotation, rib hump, kyphosis, lordosis, orientation of the plane of maximum deformity, and etc.).

2.1.1.2 Characteristics

We received 41 data from Sainte Justine Hospital at the beginning. The internal deformity of the scoliotic spine was characterized by the clinical Cobb angle. The Cobb angle of right-oriented spine (viewing from the back of the patient) was defined as positive; the Cobb angle of left-oriented spine was defined as negative. There are maximally 120 points per wire, and maximally 12 wires per patients (depends on the height of the patients). Some data points on each wire may be missing. The reason was various, for instance, the mat could not be 100 percent tightened onto the patient's trunk, which led to that some sensors did not get data. The sensor wire was divided as left half and right half. In a few extreme cases, the whole right half part of the sensor wire was missing; hence we had completely no information about the surface part under the missing sensor wire. The characteristics of these brace data were summarized in the following table.

No.	Wire numbers	Cobb angle (PA2)	King Class	Class Label	No. of PA	Thoracic	Thoracical lumbar	Lumbar	Apex
1	11 right	7.8	K2	-1	4	7.8		-6	T6
2	10 right	32.1	K1	1	3	32.1	-34.2		L1
3	11 right	13.6	K1	-1	3	13.6		-17.8	L2
4	12 right	18.8(?)	K1	-1	5	18.8		-18.9	L2
5	12 right	36.0	K2	1	3	36		-24	T9
6	10 right	40.0	K3 or K2	1	4	40	-24.1		T8
7	8 right	30.7	K2 or K4	1	3		30.7	-30.1	T12
8	12 right	41.8	K2	1	3	41.8		-31.4	T9
9	9 right	25.9	K3 or K2	-1	3	25.8	-14.4		T8
10	12 right	23.5(?)	K1	-1	5	23.5	-32.3		L1
11	9 right	39.6	K2	1	3	39.6		-35	T8
12	10 right	35.7	K2	1	3	35.7		-34.8	T10
13	10 right	19.1	K1	-1	3	19.1		-33.3	L3
14	10 right	32.6	K2	1	3	32.6		-32.8	T9
15	9 right	32.9	K2	1	3	32.9		-29.9	T10
16	11 right	41.9	K2	1	3	41.9		-36.8	T8
17	11 right	27.5	K2	-1	4	27.5		-20.7	T8
18	12 right	4.6	K1	-1	4	4.6	-10.2		L1
19	10 right	27.9	K2	-1	3	27.9		-18.6	T7
20	10 right	14.7	K1	-1	4	14.7		-29.7	L2
21	10 left	23.8	K2	-1	3	23.8		-23.6	T9
22	12 right	27.1	K1	-1	3	27.1		-34.6	L2
23	12 right	44.4	K2	1	3	44.4		-37.7	T8
24	10 right	16.8	K2	-1	3	16.8		-19.7	T10
25	9 right	40.7	K2	1	3	40.7		-28.7	T10
26	12 right	23.2	K1	-1	4	23.2	-24.7		L1
27	10 right	26.1	K2	-1	3	26.1		-28.6	T9
28	9 right	46.1	K2	1	3	46.1		-42.7	T8
29	9 right	64.6	K2	1	3	64.6		-32.1	T9
30	9 right	38.5	K2	1	3	38.5		-34.6	T10
31	12 right	11.4	K2	-1	3		11.4	-8	T12
32	9 right	30.7(?)	K2	1	4	-40.2		35.7	T12
33	10 right	24.9	K1	-1	3	24.9		-32.3	L2
34	10 right	28.8	K2	-1	3	28.8		-21.6	T10
35	9 right	18.7	K1	-1	4	18.7		-23.2	L2
36	10 right	28.9	K2	-1	4	28.9		-26.5	T8
37	12 right	31.5	K2 or K3	1	3	31.5		-28.5	T8
38	11 right	29.2	K2	-1	3	29.2		-23.4	T8
39	10 right	29.9	K2	1	3		29.9	-30.4	T12
40	12 right	40.1	K2	1	3	40.1		-30.2	T10
41	12 right	33.7	K2	1	3	33.7		-34.3	T7

Wire numbers = number of available sensor wires, left or right indicates that the last wire contains only the left half (left) or both left and right half (right). Cobb angle (PA2) = the second Cobb angle calculated from the spine from PA (posterior-anterior) view. King class = to which category according to King's classification of scoliotic spine the shape of the spine belongs (The explanation of King's classification will be given in later sections). Class label = to which class each patient's spine was assigned according to the classification criteria (given in later sections). No. of PA: number of Cobb angles calculated from the spine from PA view. Thoracic = Cobb angle at thoracic part. Thoracic lumbar = Cobb angle at thoracic lumbar part. Lumbar = Cobb angle at lumbar part. Apex = apex location.

Table 2.1: Characteristics of raw brace data

2.1.2 Raw Calgary Data

2.1.2.1 Acquisition Technique

The laser imaging system (jointly developed by the National Research Council of Canada, The Alberta Research Council, and Clynch Technologies Inc., Calgary) consisted of four BIRIS laser scanners mounted on a mobile ring and connected to a computer camera-control and data acquisition system (Figure 2.3). The four laser scanners were placed around the patient. Each of them captured 3D coordinates of parts of the patient's torso-surface by projecting a low-power (15mW) laser beam onto the torso, scanning points in sequence along each horizontal row. A customized package of computer programs was written to transform raw 3D points from four torso surface scanners into a 360° surface model. Some post-processing operations, e.g., rectification, registration, and cleansing of spurious points, were carried out too. Detailed description about the laser scanned data acquisition technique can be found in (Jaremko, 2001). Once registered in the same coordinate system the images were merged together. Contour lines were then created using surface interpolation. All interpolated points were computed from contour lines. The data we received from Calgary University was the interpolated data instead of the raw scanned data. But in order to distinguish from the data which was eventually sent into the SVM, we still gave it the name of *raw Calgary data*. Basically speaking, these raw Calgary data were very similar as the raw brace data. Taking out of the factor of different ways of acquisition, the only difference was that the raw Calgary data was more accurate in representing the surface shape of the patient than the raw brace data.

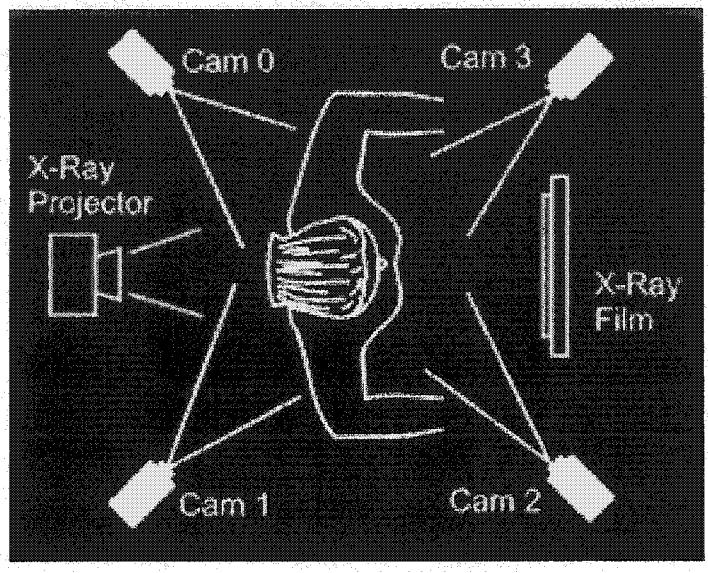


Figure 2.3: Illustration of synchronous laser scan and X-ray process (Jaremko, 2001)

2.1.2.2 Characteristics

We received 115 patients' data in all from Calgary. Data collection was done once per six months between May 1998 and May 2000. Usually the same range (75 cm) of zone of the torso of patient was scanned no matter the patient size but then the part below the PSIS dimples and above T1 (or C7) was cut. Thus, Small patients had less contour lines. The range of number of contour lines varied from 31 to 47. Each contour line contained exactly 360 points. Therefore, the scanned number of points for each patient varied from 11,160 to 16,920. Since all points on the same contour line were interpolated from the contour line, they were all computed to get the same Y (up direction) coordinates.

Arm removal operation was done to each patient. During the scanning and X-ray procedure, each patient stood in the positioning frame with arms raised to shoulder height (figure 2.4). Due to asymmetric arm positioning and non-horizontal camera orientation, contours in the upper thoracic region often captured more of one arm of the patient than the other, causing errors in calculation of left-right asymmetry indices (used in Jaremko's method). So arm removal operation was taken. This caused that some data points on the

contour lines crossing the arm part were missing, and thus these contour lines contained less than 360 points. In order to get 'rectangular' data point matrix, the removed points were replaced by NaN (Not a Number) value in the data matrix so there was always 360 components for each contour line.

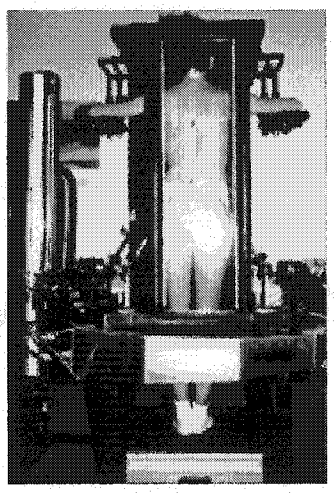


Figure 2.4: Positioning device for scanning and X-rayed (Poncet et al., 2000a)

Similarly to the Brace data, right-deformed spine (from PA view) was defined as having positive Cobb angle, left-deformed spine was defined as having negative Cobb angle. Four Cobb angles were involved in our experiments, they are Cobb angle of primary curve (mclincobb1), absolute value of the Cobb angle of primary curve (mclincobb1abs), Computer-Cobb angle of curve best matched to primary curve from chart (mmtlcobb1), absolute value of computer-Cobb angle of curve best matched to primary curve from chart (mmtlcobb1abs), respectively. mclincobb1 and mclincobb1abs are clinical Cobb angle and were calculated from curve chart of scoliotic spine by clinician. mmtlcobb1 and mmtlcobb1abs are computer Cobb angle and were generated at Ste-Justine Hospital along with the 3D spine reconstruction. There were 24 data with negative Cobb angle, the other 91 data with positive Cobb angle. The magnitude range of these Cobb angle varied from -57 degree to 75 degree. The histograms of these four Cobb angle are given out in the below (X-axis represents Cobb angle, Y-axis represents the number of patients):

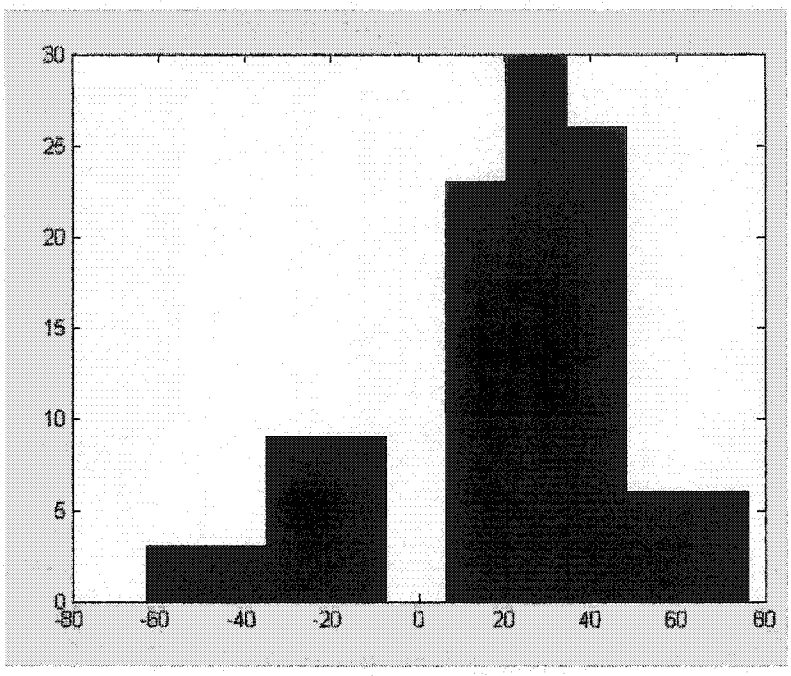


Figure 2.5: Histogram of mclincobb1

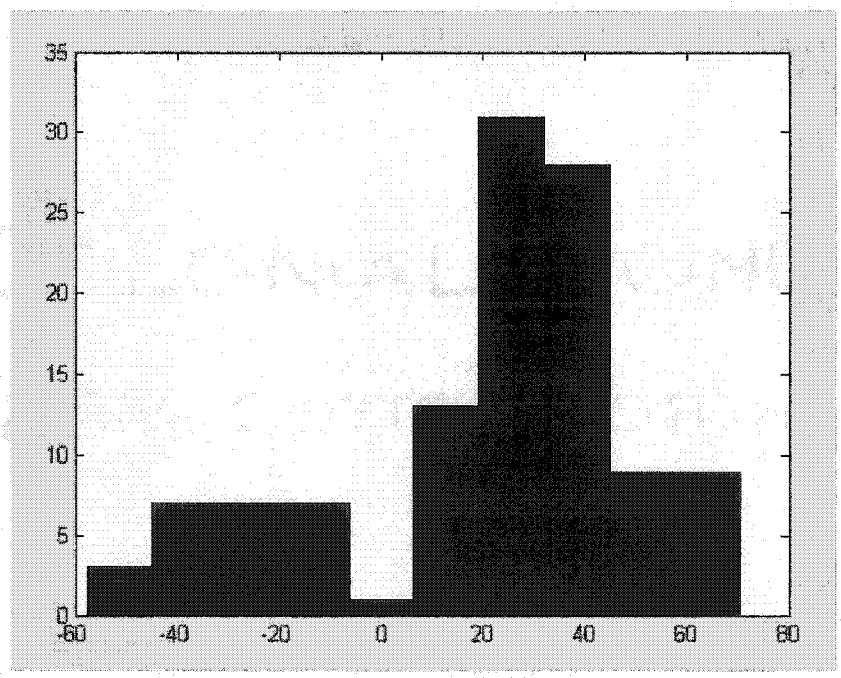


Figure 2.6: Histogram of mmtlcobb1

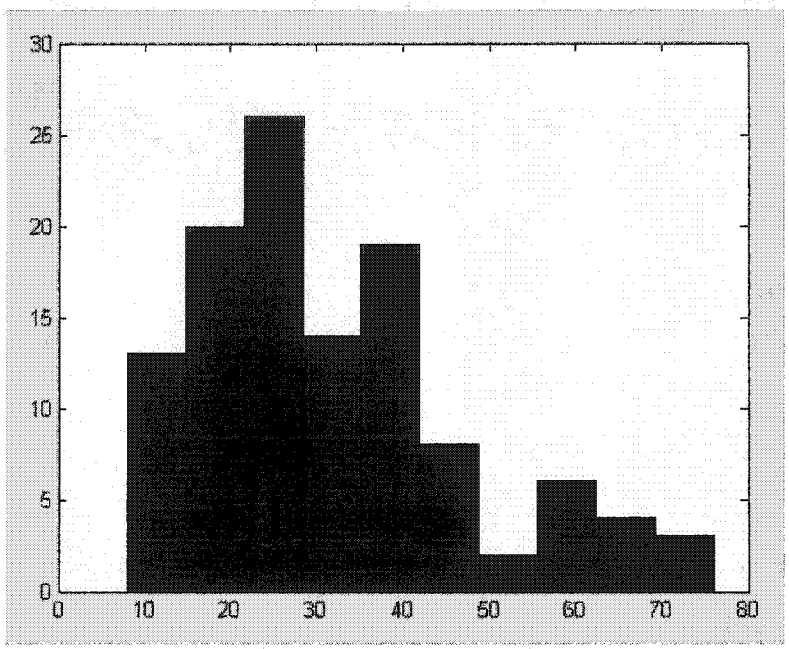


Figure 2.7: Histogram of mclincobb1abs

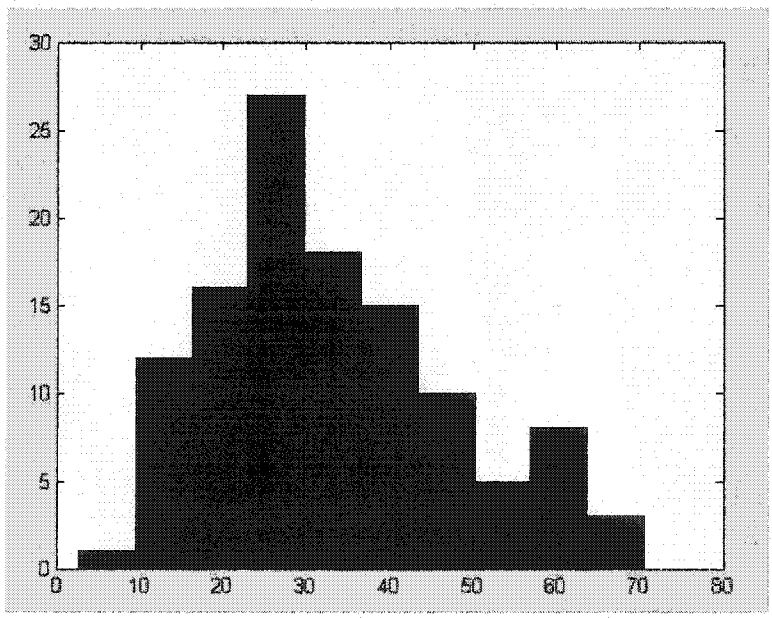


Figure 2.8: Histogram of mmtlcobb1abs

From these histograms we can see that most patients had a Cobb angle in the range of 10 ~ 40 degrees. The histogram of clinical Cobb angle and the histogram of computer Cobb angle were very similar. This feature implied the strong correlation between clinical Cobb angle and computer Cobb angle.

2.2 Surface Fitting

Theoretically speaking, we can simply send all of these data points into the learning machine to estimate the deformation of the spine if the machine had unlimited computational power. But due to the practical difficulties, manipulating these hundreds of thousands of point-cloud data is difficult and tedious. For instance, the Calgary data we had can have up to 16,920 points, and these points are 3D points which means they have 3 coordinates (in X, Y, and Z direction, respectively), thus the dimension of the Calgary data can go up to about 50,000. In fact, the laser imaging system of Calgary can capture as many as 65,536 points if higher density is required (Jaremko, 2001). We have to look for a compact representation of these vast rows of digitized 3D points. Since our starting point was to avoid doing feature extraction, our approach was to fit a smooth and continuous surface to these geometrical points, then the set of the control points of this surface could be used to represent the original data points which honestly represent the spatial structure of the torso of the patient.

2.2.1 Fitting Method Selection

Fitting techniques deal with the problem of using a few data items, e.g., coefficients, weights, to uniquely specify the entire object. These objects are usually represented by a dense set of sampling. Logically, we can rewrite this problem as:

Input: 3D data points

Process: fitting techniques

Output: curve or surface, i.e., the control points and the knots of this obtained curve or surface.

In order to choose an appropriate fitting algorithm for our problem, we had to deal with several considerations. The first consideration is about interpolation or approximation fitting. There are two types of fitting techniques: interpolation and approximation. In interpolation, the constructed curve or surface satisfies the given data precisely, i.e., the curve or surface passes through all of data points. In approximation, the constructed curve or surface usually only approximates those data points under a certain norm, e.g., the least square minimization, instead of passing through them precisely. Interpolation is usually used in theoretical analysis and in the application of few data points. In real-world applications, since the data points collected via a digitizing process can contain measurement or computational noise, and usually their number is huge, it is almost impossible and impractical to "wiggle" a curve or surface through all of the data. It is more practical to find a curve or surface which best fits the data. Another consideration is about global or local algorithm. Theoretically, in the curve or surface constructed by the global algorithm, a change to any one input data point can change the shape of the entire curve or surface, but the magnitude of the change decreases while the affected data point keeps "fleeing" away. Local algorithms construct curve or surface in a segment-wise fashion, using only local data for each step. Consequently, a change to a data point has only local effect on the entire curve or surface. However, achieving desired levels of continuity at the joint of segments is a headache, and local algorithms often result in multiple interior knots. A nice property about global algorithms is that when the degree, knots, and weights have been pre-selected, and the control points are the only unknowns, then the system of equations is linear and easy to solve.

Based on the above two considerations, we decided to adopt the global least squared approximation method (Piegl and Tiller, 2000). A degree (p, q) NURBS (Non Uniform Rational B-Splines) surface was sought to approximate the data. The advantage of this algorithm is that it can handle data points of arbitrary topology other than the traditional *"rectangular point carpet"* one.

2.2.2 NURBS Notations

For convenience, we introduce NURBS curve and surface definitions first.

- A pth-degree NURBS curve: $C(u) = \dfrac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \quad a \le u \le b$

- A NURBS surface of degree p in the u direction and degree q in the v direction:

$$S(u,v) = \frac{\sum_{i=1}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_i P_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_i} \quad 0 \le u, v \le 1$$

Where P_i are the control points, and N are the base functions.

2.2.3 Least Square Approximation

A central ingredient in the approximation method is that given a fixed number of control points, say n, we fit an approximating curve (surface) to the data. There are many ways to do this. For example, a nonlinear optimization problem can be set up, with the control points, knots, or the weights as unknowns. The objective is to minimize the error in some way, e.g., least squares or maximum deviation (see Laurent-Gengoux et al., 1993). At this part, we present a curve and surface fitting method in which a fixed number of control points are the only unknowns, and they are solved using least squares technique. We seek a *pth* degree non-rational curve

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \quad u \in [0,1]$$

satisfying that:

- $Q_0 = C(0)$ and $Q_m = C(1)$;
- the remaining Q_k are approximated in the least squares sense, i.e.

$$\sum_{k=1}^{m-1} \left\| Q_k - C(\bar{u}_k) \right\|^2$$

is a minimum with respect to the $n+1$ variables, P_i; the $\{\bar{u}_k\}$ are the pre-computed parameter values; the Q are the actual data points. We emphasize that the resulting curve generally does not pass precisely through Q_k, and $C(\bar{u}_k)$ is not the closest point on $C(u)$ to Q_k. Let

$$R_k = Q_k - N_{0,p}(\bar{u}_k)Q_0 - N_{n,p}(\bar{u}_k)Q_m \quad k = 1,\cdots,m-1$$

Then

$$f = \sum_{k=1}^{m-1} |Q_k - C(\bar{u}_k)|^2 = \sum_{k=1}^{m-1} \left| R_k - \sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k)P_i \right|^2$$
$$= \sum_{k=1}^{m-1} \left[R_k \cdot R_k - 2\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k)(R_k \cdot P_i) + \left(\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k)P_i \right) \cdot \left(\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k)P_i \right) \right]$$

f is a scalar-valued function of the $n-1$ variables, $P_1,\cdots,P_{n-1}$. Now we apply the standard technique of linear least squares fitting to minimize f we set the derivatives of f with respect to the $n-1$ points, P_l, equal to zero. The *lth* derivative is

$$\frac{\partial f}{\partial P_l} = \sum_{k=1}^{m-1} \left(-2N_{l,p}(\bar{u}_k)R_k + 2N_{l,p}(\bar{u}_k)\sum_{i=1}^{n-1} N_{i,p}(\bar{u}_k)P_i \right)$$

which implies that

$$-\sum_{k=1}^{m-1} \left(N_{l,p}(\bar{u}_k)R_k + \sum_{i=1}^{n-1}\sum_{i=1}^{n-1} N_{l,p}(\bar{u}_k)N_{i,p}(\bar{u}_k)P_i \right) = 0$$

It follows that

$$\sum_{i=1}^{n-1} \left(\sum_{i=1}^{n-1} N_{l,p}(\bar{u}_k)N_{i,p}(\bar{u}_k) \right) P_i = \sum_{k=1}^{m-1} N_{l,p}(\bar{u}_k)R_k$$

Letting $l = 1,\cdots,n-1$ yields the system of $n-1$ equations in $n-1$ unknowns $(N^T N)P = R$

where N is the $(m-1)\times(n-1)$ matrix of scalars

$$N = \begin{bmatrix} N_{1,p}(\bar{u}_1) & \cdots & N_{n-1,p}(\bar{u}_1) \\ \vdots & \ddots & \vdots \\ N_{1,p}(\bar{u}_{m-1}) & \cdots & N_{n-1,p}(\bar{u}_{m-1}) \end{bmatrix}$$

R is the vector of $n-1$ points

$$R = \begin{bmatrix} N_{1,p}(\overline{u}_1)R_1 + & \cdots & +N_{1,p}(\overline{u}_{m-1})R_{m-1} \\ & & \vdots \\ N_{n-1,p}(\overline{u}_1)R_1 & \cdots & +N_{n-1,p}(\overline{u}_{m-1})R_{m-1} \end{bmatrix} \text{ and } P = \begin{bmatrix} P_1 \\ \vdots \\ P_{n-1} \end{bmatrix}$$

The solution of this equation gives the desired control points.

The surface approximation scheme, builds upon our least squares curve scheme, is very simple, but is quite adequate for most applications. We simply fit curves across the data in one direction, and then fit curves through the resulting control points across the other direction. A comprehensive discussion about NURBS and curve and surface fitting techniques can be found in (Piegl and Tiller, 1994).

2.2.4 Algorithms

The two algorithms given below were the algorithms used in our fitting process to the raw data points with the least square approximation method.

2.2.4.1 Least Square Curve Approximation

The main idea of this algorithm is given out as below:

Input:

- 3D coordinates of data points $Q_i \quad i = 0,...,k$
- Curve degree: p
- Number of control points: n
- Set the weights to 1, in order to avoid the nonlinear problem
- Constrain matrix of data points
- Derivative matrix of data points
- Constrain matrix of the derivatives

Satisfying:

- Minimize $\sum_{i=1}^{k-1} |Q_i - C(t_i)|^2$ with respect to the control points $P_1,...,P_{n-1}$.

Output:

- The 3D coordinates of those *n+1* control points of the curve which approximates those data points in the least square sense.
- Knot vector U

The pseudocode of this algorithm is given in figure 2.9:

```
WCLeastSquaresCurve(Q,r,Wq,D,s,I,Wd,n,p,U,P)
{  /*  Weighted & constrained least squares curve fit  */
   /*  Input:  Q, r,Wq,D,s,I,Wd,n,p  */
   /*  Output:  U,P  */
   ru = -1;          rc = -1;
   for (i=0; i<=r; i++)
 if (Wq[i] > 0)  ru =ru+1;
 else  rc = rc+1;
su = -1;   sc = -1;
for (j=0; j<=s; j++)
 if (Wd[j] > 0)   su = su+1;
 else  sc = sc+1;
mu = ru+su+1;   mc = rc+sc+1;
if (mc >= n || mc+n >= mu+1)   return(error);
Compute and load parameters u_k  into ub[];
Compute and load the knots into U[];
   /*  Now set up arrays N,W,S,T,M  */
j = 0;      /*  current index into I[]  */
mu2 = 0;   mc2 = 0;       /*  counters up to mu and mc  */
for (i=0; i<=r; i++)
{
```

```
span = FindSpan(n,p,ub[i],U);
dflag = 0;
if (j <= s)
 if (I = = I[j])  dflag = 1;
if (dflag = = 0)          BasisFuns(span,ub[i],p,U,funs);
 else   DersBasisFuns(span,ub[i],p,1,U,funs);
if (Wq[i] > 0)
{       /*   Unconstrained point   */
  W[mu2] = Wq[i];
  Load the mu2th row of N[][] from funs[0][];
  S[mu2] = W[mu2]*Q[i];
  mu2 = mu2 +1;
}
else
{ /*   Constrained point   */
      Load the mc2th row of M[][] from funs[0][];
      T[mc2] = Q[i];
      Mc2 = mc2+1;
 }
if (dflag = = 1)
{/*   Derivative at this point   */
   if (Wd[j] > 0)
   { /*   Unconstrained derivative   */
     W[mu2] = Wd[j];
     Load the mu2th row of N[][] from funs[1][];
     S[mu2] = W[mu2]*D[j];
     Mu2 = mu2+1;
  }
```

```
        else
        { /* Constrained derivative */
            Load the mc2th row of M[][] from funs[1][];
            T[mc2] = D[j];
            Mc2 = mc2+1;
          }
          j = j+1;
        }
    } /* End of for-loop i=0,...,r */
    Compute the matrices N^T WN and N^T WS ;
    LUDecomposition( N^T WN ,n+1,p);
    If (mc < 0)
    { /* No constraints */
        Use ForwardBackward() to solve the control points P[],
        Equation N^T WNP + M^T A = N^T WS reduces to
                    (N^T WN)P = N^T WS.
        return;
    }
    Compute the inverse (N^T WN)^-1, using ForwardBackward();
    Do matrix operation to get: M(N^T WN)^-1 M^T and
                                M(N^T WN)^-1 (N^T WS) - T;
    Solve equation M(N^T WN)^-1 M^T A = M(N^T WN)^-1 N^T WS - T for
    the Lagrange multipliers, load into A[];
    Then P = (N^T WN)^-1 ((N^T WS) - M^T A);
}
```

Figure 2.9: Pseudocode of curve fitting algorithm

2.2.4.2 Least Square Surface Approximation

The main idea of this algorithm is given out as below:

Input:

- 3D coordinates of (r+1)*(s+1) set of data points $Q_{k,l} \quad k = 0,...,r \text{ and } l = 0,...,s$
- Pre-computed parameters $t_i \quad i = 0,...,k$
- Degree of u direction, p, and degree of v direction, q.
- Number of control points: $(n+1)*(m+1)$
- Set the weights to 1, in order to avoid the nonlinear problem

Satisfying:

- Interpolate the four corner points $Q_{0,0}, Q_{r,0}, Q_{0,s}, Q_{r,s}$ precisely.
- Fit curves across the data in one direction, and then fit curves through the resulting control points across the other direction. Both use the least squares curve approximation algorithm.

Output:

- The 3D coordinates of those $(n+1)*(m+1)$ control points of the surface which approximates those data points in the least square sense.
- Knot vectors U and V

The pseudocode of this algorithm is given out as below:

```
GlobalSurfApproxFixednm(r,s,Q,p,q,n,m,U,V,P)
{  /*  Global surface approx with fixed number of control points  */
   /*  Input:   r,s,Q,p,q,n,m  */
   /*  Output:  U,V,P  */
   SurfMeshParams(r,s,Q,ub,vb);
   Compute knots U by equations: d = (m+1)/(n-p+1) and
```

$$i = \text{int}(jd) \quad \alpha = jd - i$$
$$u_{p+j} = (1-\alpha)\bar{u}_{i-1} + \alpha\bar{u}_i \quad j = 1,\ldots,n-p \;;$$

Compute knots V by the same equations as above;

Compute Nu[][] and NTNu[][] using equation:

$$N = \begin{bmatrix} N_{1,p}(\bar{u}_1) & \cdots & N_{n-1,p}(\bar{u}_1) \\ \vdots & \ddots & \vdots \\ N_{1,p}(\bar{u}_{m-1}) & \cdots & N_{n-1,p}(\bar{u}_{m-1}) \end{bmatrix};$$

LUDecomposition(NTNu,n-1,p);

For (j=0; j<=s; j++)

{ /* u direction fits */

Temp[0][j] = $Q_{0,j}$; Temp[n][j] = $Q_{r,j}$;

Compute and load Ru[] with equations:

$$R_k = Q_k - N_{0,p}(\bar{u}_k)Q_0 - N_{n,p}(\bar{u}_k)Q_m \quad k = 1,\ldots,m-1$$
$$R = \begin{bmatrix} N_{1,p}(\bar{u}_1)R_1 + \cdots + N_{1,p}(\bar{u}_{m-1})R_{m-1} \\ \vdots \\ N_{n-1,p}(\bar{u}_1)R_1 + \cdots + N_{n-1,p}(\bar{u}_{m-1})R_{m-1} \end{bmatrix} \;;$$

Call ForwardBackward() to get the control points

Temp[1][j],…,Temp[n-1][j];

}

Compute Nv[][] and NTNv[][] using equation:

$$N = \begin{bmatrix} N_{1,p}(\bar{u}_1) & \cdots & N_{n-1,p}(\bar{u}_1) \\ \vdots & \ddots & \vdots \\ N_{1,p}(\bar{u}_{m-1}) & \cdots & N_{n-1,p}(\bar{u}_{m-1}) \end{bmatrix};$$

LUDecomposition(NTNv,m-1,q);

For (i=0; i<=n; i++)

{ /* v direction fits */

```
            P[i][0] = Temp[i][0];       P[i][m] = Temp[i][s];
            Compute and load Rv[] with the same equations as in computing
            Ru[];
            Call ForwardBackward() to get the control points
            P[i][1],…,P[i][m-1];
        }
    }
```

Figure 2.10: Pseudocode of surface fitting algorithm

2.3 Data Normalization

Data normalization is required for particular kernels due to their restricted domain, and may also be advantageous for unrestricted kernels. To determine if normalization (isotropic or non-isotropic) of the data is necessary, consideration of the input features is required. Additionally, normalization will improve the condition number of the Hessian in the optimization problem. In our experiments, since attributes of our data were in large ranges, and the ranges of attributes of different patients were significantly different due to the difference of size and height between patients, we committed normalization operation on all of our data sets before sending them into PCA or SVM. We merged the two datasets: training set and test set, as one and scaled it. We then split them again for training and testing. The normalization was implemented in the following manner (libsvm2.36):

```
Lower = -1;
Upper = 1;
[MaxV, I]=max(Data);
[MinV, I]=min(Data);
[R,C]= size(Data);
scaled=(Data-ones(R,1)*MinV).*(ones(R,1)*((Upper-Lower)*ones(1,C)./(MaxV-
MinV)))+Lower;
```

In the experiments, we set the normalization range of data between [-1, 1].

2.4 Principal Component Analysis

After doing the surface fitting, we obtained a compact representation of the original raw data points with the set of control points. Compared to the number of raw points, the number of control points was much less. From the point of view of using these control points instead of raw points as input to the learning machine, the dimension of the data set was drastically reduced which made the computation practical and more efficient. For instance, a raw brace data contains about 120*11=1320 points. By applying surface fitting we obtained 8*8*2=128 control points. Therefore, the dimension of each data was reduced from 1320*3= 3960 to 128*3=384 (each point was a 3D point, so we must multiply by 3). In the case of raw Calgary data, the dimension of data was reduced from 360*46*3=49680 to 41*11*3=1353. Data with this size can be directly sent into the learning machine. In fact, we did it. But based on the very limited available data (41 data in brace case, 115 data in Calgary case), we still wanted to continually reduce the dimension of data with the hope that learning machine could perform better on dataset with less dimensions. For this purpose, we investigated the method of principal component analysis (PCA).

The multivariate statistical method of PCA is a very useful tool for reducing the number of variables in a data set. In many data analysis application, we are faced with contradictory goals: On one hand, we should simplify the problem by reducing the dimension of the representation. On the other hand we want to preserve as much as possible of the original information content. PCA offers a convenient way to control the trade-off between losing information and simplifying the problem at hand. The main idea of PCA is to find an orthogonal set of basis vectors (eigenvectors) for the feature space, subject to the requirement that the new features have zero correlation with each other. First basis -- data projected on which will have the maximum variance, the second basis captures the second maximum variance that is orthogonal to the first one, and so on and so forth (figure 2.11). Once the variances are sufficiently captured by some bases, the

subsequent bases can be discarded. This helps us to achieve dimensionality reduction. Principal components are the eigenvectors of the covariance matrix of the data set.

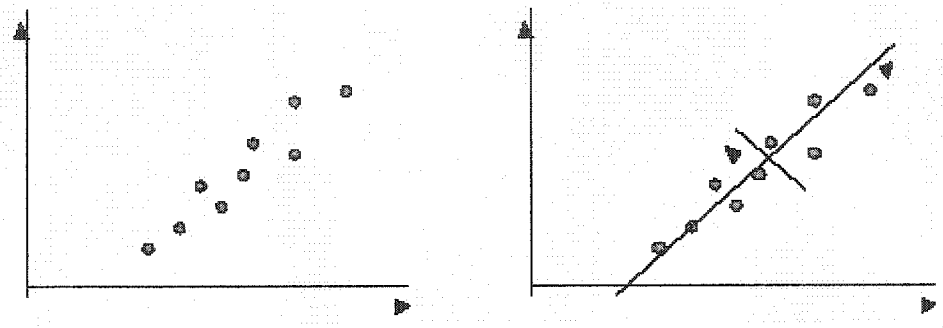


Figure 2.11: Illustration of eigenvectors of an artificially created dataset

2.4.1 Calculation of Principal Components

PCA is based on the statistical representation of a random variable. Suppose we have a random vector population x, where

$$x = (x_1,...,x_n)^T$$

and the mean of that population is denoted by

$$\mu_x = E\{x\}$$

and the covariance matrix of the same data set is

$$C_x = E\{(x-\mu_x)(x-\mu_x)^T\}$$

The components of C_x, denoted by c_{ij}, represent the covariances between the random variable components x_i and x_j. The component c_{ii} is the variance of the component x_i. The variance of a component indicates the spread of the component values around its mean value. If two components x_i and x_j of the data are uncorrelated, their covariance is zero $(c_{ij} = c_{ji} = 0)$. The covariance matrix is, by definition, always symmetric. From a sample of vectors $x_1,...,x_M$, we can calculate the sample mean and the sample covariance matrix as the estimates of the mean and the covariance matrix. From a symmetric matrix

such as the covariance matrix, we can calculate an orthogonal basis by finding its eigenvalues and eigenvectors. The eigenvectors e_i and the corresponding eigenvalues λ_i are the solutions of the equation

$$C_x e_i = \lambda_i e_i, \; i = 1,...,n$$

By ordering the eigenvectors in the order of descending eigenvalues (largest first), one can create an ordered orthogonal basis with the first eigenvector having the direction of largest variance of the data. In this way, we can find directions in which the data set has the most significant amounts of energy.

Suppose one has a data set of which the sample mean and the covariance matrix have been calculated. Let A be a matrix consisting of eigenvectors of the covariance matrix as the row vectors. By transforming a data vector x, we get

$$y = A(x - \mu_x)$$

which is a point in the orthogonal coordinate system defined by the eigenvectors. Components of y can be seen as the coordinates in the orthogonal base. We can reconstruct the original data vector x from y by

$$x = A^T y + \mu_x$$

using the property of an orthogonal matrix $A^{-1} = A^T$. A^T is the transpose of a matrix A. The original vector x was projected on the coordinate axes defined by the orthogonal basis. The original vector was then reconstructed by a linear combination of the orthogonal basis vectors.

Instead of using all the eigenvectors of the covariance matrix, we may represent the data in terms of only a few basis vectors of the orthogonal basis. If we denote the matrix having the K first eigenvectors as rows by A_K, we can create a similar transformation as seen above

$$y = A_K (x - \mu_x)$$

and

$$x = A_K^T y + \mu_x$$

This means that we project the original data vector on the coordinate axes having the dimension K and transforming the vector back by a linear combination of the basis vectors. This minimizes the mean-square error between the data and this representation with given number of eigenvectors.

If the data is concentrated in a linear subspace, this provides a way to compress data without losing much information and simplifying the representation. By picking the eigenvectors having the largest eigenvalues we lose as little information as possible in the mean-square sense. One can for instance choose a fixed number of eigenvectors and their respective eigenvalues and get a consistent representation, or abstraction of the data. This preserves a varying amount of energy of the original data. Alternatively, we can choose approximately the same amount of energy and a varying amount of eigenvectors and their respective eigenvalues. This would in turn give approximately consistent amount of information in the expense of varying representations with regard to the dimension of the subspace.

2.4.2 How Many Principal Components?

The major objective in many applications of PCA is to replace the p elements of **x** by a much smaller number, m, of PCs, which nevertheless discard little information. It is crucial to know how small m can be taken without serious information loss. Various rules have been proposed for determining a suitable value of m (Jolliffe, 1986). We chose the rule of cumulative percentage of total variation. The idea of it is to select a (cumulative) percentage of total variation, which it is desired that the selected first m PCs should contributes, say 80% or 90%. Since the PCs are organized in a descend order, the required number of PCs is then the smallest number for which this chosen percentage is exceeded.

2.5 Support Vector Machine

In fact, SVM is motivated by the consideration of training linear machines with margins, but reply on pre-processing the data to represent patterns in a high dimension – typically much higher than the original feature space. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. In this part, we will first briefly recall the mechanism and the characteristics of SVM, then describe how we applied it into our problem.

2.5.1 Optimal Separating Hyperplane

We consider only 2-dimension 2-class problem first. Given a set of points that belong to either of two classes on the plane, how to separate them according to their class labels is a crucial problem in the field of pattern recognition. There are many approaches existing currently to it. For instance, in the following illustrated example:

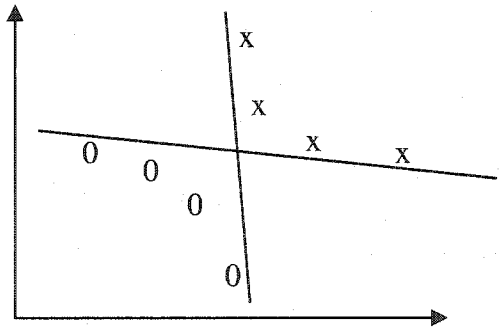


Figure 2.12: Separating hyperplane

In the above figure, the data points are linearly separable. Both separating hyperplane (the straight lines in the figures) perfectly separate these two classes, namely, the error is 0. In fact, any random straight line that passes through the zone clamped by these two lines can perfectly separate these two sets of points. Now, a question arises: which one to choose among these infinite set of separating lines? Certainly we want the best one, but a

new question is how to evaluate the performance of each separating hyperplane. Fortunately we have answer for the latter question. From the point of view of pattern recognition, the best separating hyperplane is the one who gains the best performance on the test set, i.e. the future-coming data. The currently available data is termed as the training set, which is used to train our decision function. This answer implicitly responds to the first question too, namely, we should always choose the separating hyperplane that has the best performance on the test set. At this moment, the problem turns to how to compute such kind of '*optimal*' separating hyperplane. The working principle of SVMs is to find the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. The following figure is an intuitive illustration of the idea of SVM. The dash line is the SVM solution to this data set:

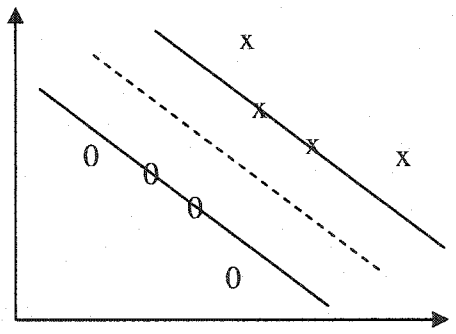


Figure 2.13: Optimal separating hyperplane

According to [Vapnik, 1995], given fixed but unknown probability distributions, this hyperplane – called Optimal Separating Hyperplane (OSH) – minimizes the risk of misclassifying not only the examples in the training set but also the *yet-to-be-seen* examples of the test set.

In the above linearly separable case, the SVM we obtained is called the linear SVM. While the data set is not linearly separable, namely, both classes of data points are mixed together. A linear separating hyperplane usually cannot get satisfying performance. At

this time we need a nonlinear SVM which can create a nonlinear decision boundary. The idea of nonlinear SVM remains simple and beautiful. It transforms the input data points using a nonlinear mapping function (termed kernel function), in other words, transforms the instance space into a new space (Hilbert space) such that the data points will be linearly separable there. With a nonlinear mapping, a straight line in the new space doesn't look straight in the original instance space. A linear model constructed in the new space can represent a nonlinear decision boundary (i.e. nonlinear SVM) in the original space. The following figure is an intuitive illustration of the idea of nonlinear SVM:

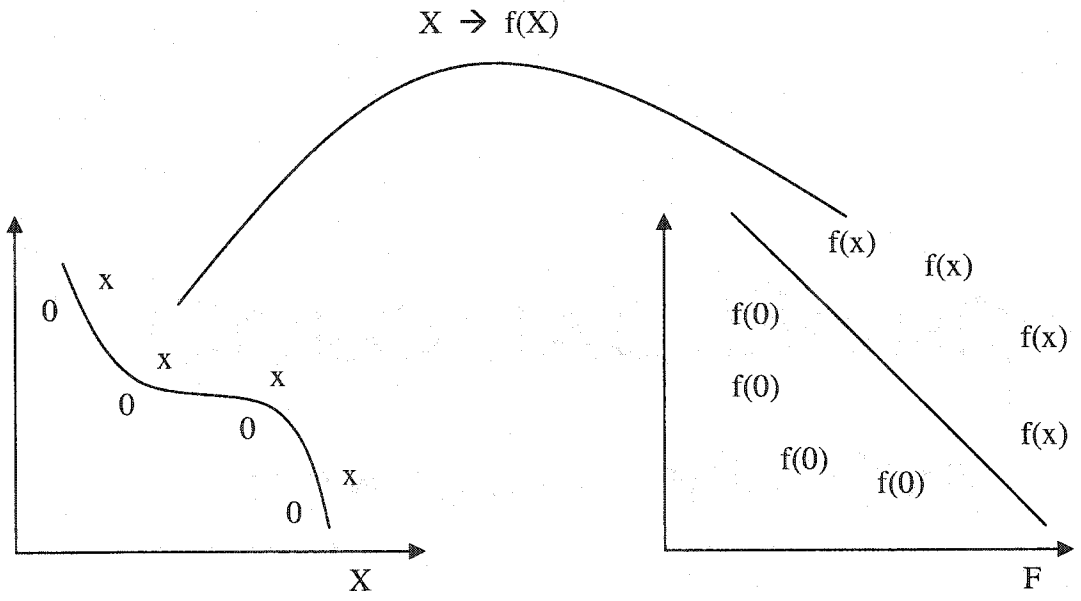


Figure 2.14: kernel function and nonlinear SVM

2.5.2 Soft Margin and Support Vectors

The OSH is called the *hard margin* classifier as well, since it searches for the hyperplane which can separate all the points of the same class on the same side, namely, it does not tolerate any mislabeled point or noise. Therefore at the presence of classification noise, the decision boundary created by a hard margin classifier will become too complex and

'*sticked*' on the training data (in order to correctly classify all training data), i.e. it overfits! The following figure illustrates this problem:

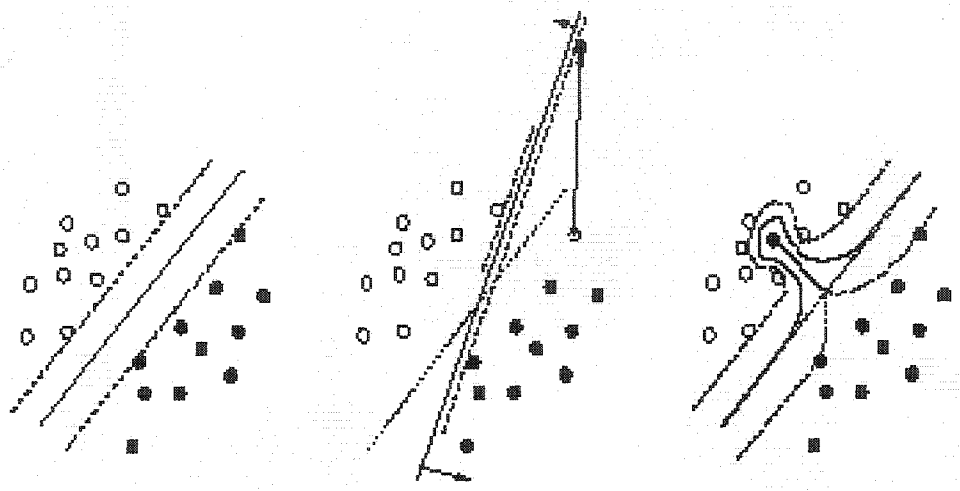


Figure 2.15: Hard margin and overfitting

The three graphs show the maximum margin hyperplane found on three datasets, respectively: on reliable data (left), on data with an outlier (middle) and on data with a mislabeled pattern (right). This figure shows that the hard margin implies noise sensitivity; only one pattern can spoil the whole estimation of the decision boundary. So the mechanism of *hard margin* classifier certainly has to be generalized to the so-called *soft margin* mechanism, which can handle with both the perfect and noisy data set, and the perfect case is only a special case of the noisy case.

2.5.3 Support Vector Classification

At this part we give out the algorithm of the linear SVM for binary non-separable classification case. The binary separable case is only its special case. We assume we are given a set of points $x_i \in R^n$ with $i = 1, 2, \ldots, N$. Each point x_i belongs to either of two classes and this is given a label $y_i \in \{-1, +1\}$. The goal is to establish the equation of a

hyperplane that divides the points leaving all the points of the same class on the same side while maximizing the distance between the two classes and the hyperplane in the *soft margin* sense. Recall that the equation of the separating hyperplane can be written as $w \cdot x + b = 0$ (see following figure):

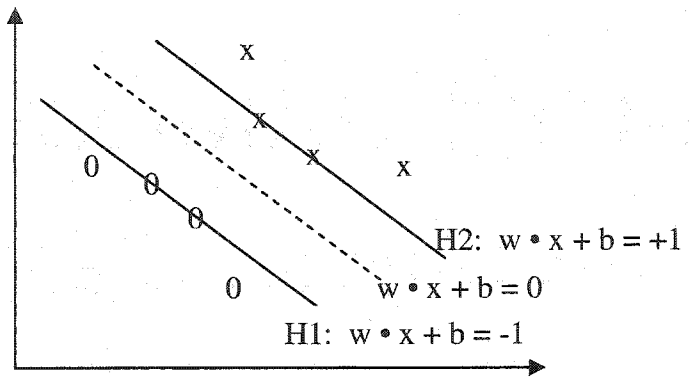


Figure 2.16: Separating hyperplane and margin

H1 and H2 are the furthest hyperplane parallel to the separating hyperplane. They are called the margin plane. The separating hyperplane locates in the middle of H1 and H2. So the equation of the margin plane can be considered as $w \cdot x + b = -1$ and $w \cdot x + b = +1$. We allow for noise, or imperfect separation. That is, we do not strictly enforce that there be no data points between H_1 and H_2, but we definitely want to penalize the data points that cross the boundaries. The penalty C will be finite (If $C = \infty$, we come back to the separable case). We introduce non-negative slack variables $\zeta_i \geq 0$ to deal with the noise. The purpose of the variables ζ_i is to allow for a small number of misclassified points. If the data points are linearly separable, then ζ_i is null. Then the margin plane H1 and H2 become:

$$w \cdot x_i + b \geq +1 - \zeta_i \quad for\ y_i = +1,$$

$$w \cdot x_i + b \leq -1 + \zeta_i \quad for\ y_i = -1,$$

$$\zeta_i \geq 0, \quad \forall i.$$

and we add to the objective function a penalizing term:

$$\underset{w,b,\zeta}{\text{minimize}} \frac{1}{2} w^T w + C(\sum_i \zeta_i)^m$$

where m is usually set to 1, which gives us

$$\begin{aligned} \underset{w,b,\zeta_i}{\text{minimize}} \quad & \frac{1}{2} w^T w + C(\textstyle\sum_{i=1}^N \zeta_i) \\ \text{subject to} \quad & y_i(w^T x_i - b) + \zeta_i - 1 \geq 0, \quad 1 \leq i \leq N \\ & \zeta_i \geq 0, \qquad\qquad 1 \leq i \leq N \end{aligned}$$

Introducing Lagrange multipliers α, β, the Lagrangian is:

$$\begin{aligned} \ell(w,b,\zeta_i;\alpha,\beta) &= \frac{1}{2} w^T w + C\sum_{i=1}^N \zeta_i \\ &\quad - \sum_{i=1}^N \alpha_i [y_i(w^T x_i - b) + \zeta_i - 1] - \sum_{i=1}^N \mu_i \zeta_i \\ &= \frac{1}{2} w^T w + \sum_{i=1}^N (C - \alpha_i - \mu_i)\zeta_i \\ &\quad - \left(\sum_{i=1}^N \alpha_i y_i x_i^T\right) w - \left(\sum_{i=1}^N \alpha_i y_i\right) b + \sum_{i=1}^N \alpha_i \end{aligned}$$

Neither the ζ_i's, nor their Lagrange multipliers appear in the Wolfe dual problem:

$$\underset{\alpha}{\text{maximize}} \ell_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

subject to:

$$0 \leq \alpha_i \leq C,$$

$$\sum_i \alpha_i y_i = 0.$$

The only difference from the perfectly separating case is that α_i is now bounded above by C instead of ∞. The solution is again given by

$$w = \sum_{i=1}^N \alpha_i y_i x_i$$

To train the SVM, we search through the feasible region of the dual problem and maximize the objective function. The optimal solution can be checked using the Karush-Kuhn-Tucker (KKT) conditions. Most of the α_i are usually null, therefore the vector w is a linear combination of a relatively small percentage of the original points set. These points are termed *support vectors* because they are the closest points from the separating hyperplane and the only points of the original points set needed to determine the separating hyperplane. All support vectors locate on the margin planes. Given a support vector x_j, the parameter b can be obtained from the corresponding KKT condition as

$$b = y_j - w \cdot x_j$$

The problem of classifying a new data point x is now simply solved by looking at the sigh of

$$w \cdot x + b$$

Therefore, the support vectors condense all the information contained in the training set which is needed to classify new data points.

2.5.4 Multi-Class Classification

SVM was originally designed for two-class classification. In order to extend it to process multi-class problem, several methods have been proposed so far where typically we obtained a multi-class SVM by combining several binary SVM. Some methods which consider all classes at once have also been proposed. Based on the fact that multi-class SVM is still an on-going research issue, we introduce only two popular methods here which are based on binary SVM combination: "one-against-all" and "one-against-one".

- **One-against-all**

The main idea of "one-against-all" is that it constructs k SVM solutions where k is the number of classes. The original dataset used to construct the ith SVM is modified by that all of the examples in the ith class are labeled positive, and all other examples are labeled

negative. Thus given l training data $(x_1, y_1),\ldots,(x_l, y_l)$, where $x_i \in R^n$, $i = 1,\ldots,l$ and $y_i \in \{1,\ldots,k\}$is the class of x_i, the ith SVM solves the following problem:

$$\begin{aligned}\min_{w_i,b_i,\zeta_i} \quad & \frac{1}{2}(w_i)^T w_i + C\sum_{j=1}^{l}\zeta_j^i \\ & (w_i)^T\phi(x_j) + b_i \geq 1-\zeta_j^i, \quad if\ y_j = i, \\ & (w_i)^T\phi(x_j) + b_i \leq -1+\zeta_j^i, \quad if\ y_j \neq i, \\ & \zeta_j^i \geq 0, \quad j = 1,\ldots,l\end{aligned}$$

where ϕ is the kernel function which maps the training data into a higher dimensional feature space. C is the penalty parameter. After solving the above problem, there are k decision functions:

$$\begin{aligned}&(w_1)^T\phi(x) + b_1, \\ &\vdots \\ &(w_k)^T\phi(x) + b_k.\end{aligned}$$

Then, a new example x is in the class which has the largest value of the k decision function:

$$\text{class of } x \equiv \arg\max_{i=1,\ldots,k}((w_i)^T\phi(x) + b_i).$$

- **One-against-one**

The "one-against-one" method computes the optimal separating hyperplane (OSH) associated to each pair of classes i and j, and store them. Thus $k(k-1)/2$ SVM solutions where each one is trained on data from only two classes are constructed. For training data from the ith and the jth classes, we solve the following binary classification problem:

$$\begin{aligned}\min_{w_{ij},b_{ij},\zeta_{ij}} \quad & \frac{1}{2}(w_{ij})^T w_{ij} + C\sum_{t}\zeta_t^{ij} \\ & (w_{ij})^T\phi(x_t) + b_{ij} \geq 1-\zeta_t^{ij}, \quad if\ y_t = i, \\ & (w_{ij})^T\phi(x_t) + b_{ij} \leq -1+\zeta_t^{ij}, \quad if\ y_t = j, \\ & \zeta_t^{ij} \geq 0.\end{aligned}$$

There are different approaches for doing the future testing. One way is to employ the rules of tennis tournament to deal with the testing. Each class is regarded as a player; all players are paired off to play matches. In each match the system temporarily classifies a test data as in a class according to the OSH relative to the pair of players involved in the match. At each round the half losing players are out, and the half winners advance to next round. Repeating the previous procedure until the final round, then the test data is classified as the champion's class.

- **Which one is better?**

Past research shows that "one-against-one" method is more suitable for practical use than the other methods (Hsu and Lin, 2002). So in this project, we adopted "one-against-one" method.

2.5.5 Kernel Functions

SVM constructs a mapping into a high dimensional feature space by the use of reproducing kernels. The idea of the kernel function is to enable operations to be performed in the input space rather than the potentially high dimensional feature space. Hence the inner product does not need to be evaluated in the feature space. This provides a way of addressing the curse of dimensionality. However, the computation is still critically dependent upon the number of training patterns and to provide a good data distribution for a high dimensional problem will generally require a large training set.

Briefly speaking, an inner product in feature space has an equivalent kernel in input space, $K(x, y) = k(x) \cdot k(y)$, provided certain conditions hold. If K is a symmetric positive definite function, which satisfied Mercer's Conditions,

$$K(x, y) = \sum_{m=1}^{\infty} \alpha_m \psi(x)\psi(y), \quad \alpha_m \geq 0$$

$$\iint K(x, y)g(x)g(y)dxdy > 0, \quad \int g^2(x)dx < \infty$$

then the kernel represents a legitimate inner product in feature space. We utilized linear, polynomial, Gaussian radial basis function, exponential radial basis function, and sigmoid in our experiments. We give out their function in the below.

2.5.5.1 Polynomial

A polynomial mapping is a popular method for non-linear modeling,

$$K(x, y) = (x \cdot y)^d \quad or \quad K(x, y) = (x \cdot y + 1)^d, \quad d = 1,.....$$

The second kernel is usually preferable as it avoids problems with the Hessian becoming zero. When d=1, the polynomial kernel becomes linear kernel.

2.5.5.2 Gaussian Radial Basis Function

Radial basis functions have received significant attention, most commonly with a Gaussian of the form,

$$K(x, y) = \exp(-\frac{(x-y)^2}{2\sigma^2})$$

2.5.5.3 Exponential Radial Basis Function

A radial basis function of the form,

$$K(x, y) = \exp(-\frac{|x-y|}{2\sigma^2})$$

produces a piecewise linear solution which can be attractive when discontinuities are acceptable.

2.5.5.4 Kernel Selection

The obvious question that arises with kernels is that with so many different kernels to choose from, which one is the best choice for our problem? The upper bound on the VC

dimension is a potential avenue to provide a means of comparing the kernels. However, it requires the estimation of the radius of the hypersphere enclosing the data in the non-linear feature space. Even if a strong theoretical method for selecting a kernel is developed, unless this can be validated using independent test sets on a large number of problems, methods such as bootstrapping and cross-validation will remain the preferred method for kernel selection. Due to the lack of theoretical analysis of the characteristics of the scoliosis datasets, we simply test the above-mentioned kernels, i.e. linear, polynomial, RBF and ERBF kernel, with cross-validation method to check out which one worked the best on the scoliosis datasets. These kernels are the most popular choices in common SVM classification applications.

2.6 Training and Testing Criteria

What we are interested in a classifier is its likely future performance on new data, not the past performance on old data. Error rate on the training set is not likely to be a good indicator of future performance, since the classifier has been learned from the very same training data, any estimate of performance based on that data will be optimistic, and may be hopelessly optimistic. So it is very important to set down the appropriate criteria for the training and testing of a classifier beforehand.

If lots of data are available, it will be easy to determine the training and testing criteria: we take a large sample and use it for training, then another independent large sample of different data and use it for testing. The commonly used way in ML community is to leave one third of total available data as the testing data, and the left two third as the training data. Provided both samples are representative, the error rate on the test set will give a true indication of future performance. Generally speaking, the larger the training sample the better the classifier, although the returns begin to diminish once a certain volume of training data is exceeded. And the larger the test sample, the more accurate the error estimate.

The real problem occurs when there is not a vast supply of data available. Unfortunately our case falls into this range. Our data is extremely limited. The problem becomes how to make the most of a limited dataset. There is a dilemma here: to get a good classifier, we want to use as much of the data as possible for training; to get a good error estimate, we want to use as much of it as possible for testing. The widely used methods for dealing with this dilemma are cross validation and leave-one-out.

2.6.1 Cross Validation

In practical terms, it is common to hold one-third of the data out for testing and use the remaining two-thirds for training. In the case that we have sufficient data at hand, this criterion works well. But in the case of limited data, we may be unlucky: the sample used for training (or testing) may not be representative. If, by bad luck, all examples with a certain class were missed out of the training set, we could hardly expect a classifier learned from that data to perform well on the examples of that class – and the situation would be exacerbated by the fact that the class would necessarily be over-represented in the test set since none of its instances made it into the training set! Instead, we should ensure that the random sampling is done in such a way as to guarantee that each class is properly represented in both training and testing sets. For doing this, a technique called cross validation has been invented. In k-fold cross-validation, we divide the data into k subsets of (approximately) equal size. We train the classifier k times, each time leaving out one of the subsets from training, but using only the omitted subset for testing. The final result is averaged over k invokes.

Now a new question appears: how to decide the fold number k? The standard way is to use 10. Why ten? Extensive tests on numerous different datasets, with different learning techniques, have shown that ten is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up. Although debate continues in machine learning circle about what is the best scheme for evaluation, 10-fold cross validation has become the standard method in practical terms.

A single 10-fold cross validation might not be enough to get a reliable error estimate. Different 10-fold cross validation experiments with the same learning scheme and dataset often produce different results, because of the effect of random variation in choosing the folds themselves. When going for an accurate error estimate, it is standard procedure to repeat the cross validation process ten times – that is, ten 10-fold cross validations – and average the results. This involves invoking the learning algorithm one hundred times, on datasets that are all nine-tenths the size of the original one. Getting a good measure of performance is a computation-intensive undertaking.

2.6.2 Leave-One-Out

10-fold cross validation is the standard way of measuring the error rate of a learning scheme on a particular dataset; for reliable results, ten times 10-fold cross validation. But many other methods are used instead. Leave-one-out is particularly prevalent when the size of dataset is extremely small. Leave-one-out is simply *n*-fold cross validation, where n is the number of instances in the dataset. Each instance in turn is left out, and the learning scheme is trained on all the remaining instances. It is judged by its correctness on the remaining instance, 1 or 0 for success or failure. The results of all *n* judgments, one for each member of the dataset, are averaged, and that average represents the final error estimate.

This procedure is an attractive one for two reasons. First, the greatest possible amount of data is used for training in each case, which presumably increases the chance that the classifier is an accurate one. Second, the procedure is deterministic: no random sampling is involved. There is no point in repeating it ten times, or repeating it all: the same result will be obtained each time.

2.7 Parameter Tuning

To obtain a good performance, some parameters in SVM have to be chosen carefully. These parameters include:

- The regularization parameter C, which determines the tradeoff between minimizing the training error and minimizing model complexity;
- Parameter gamma or d of the kernel function that implicitly defines the nonlinear mapping from input space to some high dimensional feature space.

Different setting of these parameters can cause significant difference in performance. Therefore, choosing optimal hyperparameter values for SVM is an important step in SVM design. Some works have been done on this subject during the past few years (Keerthi, 2001; Duan et al, 2001). Tuning these hyperparameters is usually done by minimizing the estimated generalization error such as the k-fold cross-validation error or the leave-one-out error, or some other related performance measure. In our experiments, a simple grid search was carried out by minimizing the estimated generalization error with cross-validation method for the optimal C and gamma on the datasets.

2.8 Data Sets

2.8.1 Class Labelling Criterion

After having obtained the data sets of control points which are the approximation to the raw brace or scanned data points, we have to assign a class label to each set of control points. The control points set of a patient corresponds to one data in the data set. How to decide the value of the class label depends on the spinal characteristics of patient since the purpose of our experiments is to correlate the external surface shape to the internal deformed spine. King's classification criterion of the spine is well accepted in clinics, but a main issue about it is that it classifies the scoliotic spine based only on the shape of the spine, in other words, it does not take into account the magnitude of the deformity of the

spine, which is more concerned by clinicians. So a magnitude concerned classification criterion should be a more appropriate choice in labeling the class of each data than King's criterion. Basing on the fact that Cobb angle is the in-fact gold standard in clinic in determining the severity of deformity of the spine, we utilized it as the class labeling criterion. The class label of each data is a category number (e.g., 1, 2, 3 or 1, -1) other than actual Cobb angle, but the determination of the category number was still based upon the actual Cobb angle of that data.

For Brace data, we had only 41 data with positive clinical Cobb angle. There were still 5 other data with negative Cobb angle, but since their number was too low to set a class for them. We simply discard these 5 data. The average Cobb angle of these 41 patients = 29.4°. For the purpose of feasibility test, we decided to separate these data into two classes. Using this average Cobb angle as the threshold, for those whose Cobb angle were larger than it, we set them in class +1; for those whose Cobb angle were smaller than it, we set them in class -1.

For Calgary data, we had two types of Cobb angle: clinical Cobb angle and computed Cobb angle. The right-deformed spine (viewing from the back of patient) was defined as having positive Cobb angle and the left-deformed spine was defined as having negative Cobb angle. Since only a few patients in our dataset had negative Cobb angle, it is hard to set a specific category for them in our training and testing procedure. But we did not want to waste these data either since the size of our dataset was already so limited, so an alternative choice was to use the absolute value of the Cobb angle as the Cobb angle of the patients. Therefore, we had four Cobb angles for each patient: clinical Cobb angle, absolute clinical Cobb angle, computer Cobb angle, and absolute computer Cobb angle. Patients were assigned into three classes according to their Cobb angle (Cobb angle<30°, 30°-50°, and >50°, respectively). These three classes correspond to patients with mild, moderate, and severe spinal curves, respectively.

2.8.2 Characteristics

2.8.2.1 Brace Data

We fitted a surface with 8x8 control points net to each half raw Brace data (each Brace data was separated into left and right half). The degree of continuity of the surface in both u and v direction was set as 3. So for a complete surface fitting to a raw Brace data had 8x8x2 = 128 control points. Each control point has three dimensions (x, y, and z coordinates). Therefore, after doing surface fitting to these 41 raw Brace data, we obtained a new dataset of 41 data that each data was the 8x8x2 control points net to its corresponding raw data. And the dimension of each new data = 8x8x3x2 = 384. This new dataset was the one that was going to be fed into the dimension reduction tool (PCA) or into SVM directly. We called this control point dataset as Brace dataset, as opposite to the previous mentioned raw Brace dataset.

In the classification experiments, according to our class labeling criterion for Brace data, we got 21 data in class -1 (whose Cobb angle < average Cobb angle) and 20 data in class +1 (whose Cobb angle > average Cobb angle). The distributions of data in two classes were balanced.

2.8.2.2 Calgary Data

Theoretically speaking, the best way to decide how many control points are required to approximate a batch of scattered points is to compute the approximation error, and then increase the number of control points iteratively until the error is under a given threshold, say 5% or 10%. Nevertheless, the method can work only with the case of one patient, other than a batch of patients. In our case, we had 115 patients, and each patient had different size and height, namely, each patient had drastically different number of raw scanned points. In fact, the contour line number of each patient varied from 31 to 47, which corresponded to that the number of raw scanned points of each patient varied from approximately 11,160 to 16,920. Thus, it is never possible to set a global threshold for all

the patients. For example, let's say that we want the approximation error is below 5% for all the patients, then for patient A we might get a control points set of 20x15, but for patient B we might get another set of 30x20. Thus each patient might have different number of control points which makes the dimension of each data in the control point dataset becomes unequal. This will increase the complexity when processed by SVM, because SVM can handle only '*rectangular*' dataset, which means that all data must have the same dimension. As a tradeoff, we fitted surfaces with fixed number of control points to all the patients in order to make the data in the control point dataset have the same dimension. So we fitted a surface with 16x8, 41x11, and 61x31 control points net, respectively, to each raw Calgary data. These three sets of control points net correspond to loose, moderate, and tight manner of fitting, respectively. So the dimension of the control point dataset was 16x8x3 = 384, 41x11x3 = 1353, and 61x31x3 = 5673, respectively. These control point datasets were the one that was going to be fed into the dimension reduction tool (PCA) or into SVM directly. We called this control point dataset as Calgary dataset, as opposite to the previous mentioned raw Calgary dataset.

In the classification experiments, according to our class labeling criterion for Calgary data, we got 59 data in class 1 (whose Cobb angle $< 30^{o}$), 41 data in class 2 (whose Cobb angle was between $30 - 50^{o}$), and 15 data in class 3 (whose Cobb angle $> 50^{o}$). The distributions of data in three classes were highly unbalanced.

CHAPTER 3 - RESULTS

3.1 Benchmark Test

Before proceeding to the real-world scoliosis data, we committed a benchmark test firstly in order to evaluate the correctness and performance of the SVM software we utilized. The SVM software package we chose was Libsvm 2.36, a simple and easy-to-use support vector machine tool for classification. Iris Plants Database from UCI Machine Learning Repository (url of it: http://www.ics.uci.edu/~mlearn/MLRepository.html) is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other. Many researchers committed classification experiments on it with all kinds of different classifiers, and they all reported very low classification error rates (Dasarathy, 1980; Gates, 1972). Since this is an exceedingly simple domain, we did not commit on parameter tuning for SVM. We simply set C = 1, Gamma = 1, Degree = 3. Error rates were computed with 10-fold cross validation training and testing method.

Kernel	Training error	Test error
Linear	2.52%	4%
Polynomial	2.67%	5.33%
RBF	2.44%	4.67%

Table 3.1: Iris benchmark test results

Besides the Iris dataset, we tested the SVM algorithm on another benchmark dataset: Thyroid gland data, which is from UCI Machine Learning Repository too. This is a medical dataset. In (Coomans et al., 1983), this dataset was used for comparing 16 different discriminant techniques, each trying to predict the state of the thyroid gland. In (Coomans & Broeckaert, 1986), the data was used to compare different kernel density

methods. Some achieved 100% correct classification. This is also a 3-class set with 215 instances, each instance contains 5 attributes. Same setting of parameters was utilized.

Kernel	Training error	Test error
Linear	4.91%	8.71%
Polynomial	1.19%	3.9%
RBF	3.82%	7.81%

Table 3.2: New-thyroid benchmark test results

From the above two results tables, we can see that our SVM package worked properly. The results we achieved are close the results reported by other researchers.

3.2 Surface Fitting Results

In all the committed fitting experiments, the degrees of continuity in both u and v direction were set to 3, i.e., p=q=3. Degree 3 of continuity is sufficient for most applications, including ours. High degree of continuity might cause strange properties and is hard to control.

3.2.1 Fitting Results on Raw Brace Data

After testing our algorithms on those artificial *'toy'* problems, we got back to our real-world data, i.e., raw brace data. We chose a contour line that was composed of 60 points from the raw brace data set, and then tested our curve approximation algorithm on it with setting to using 7 control points. After that, we chose two sets of raw points, which represented the half torso and the complete torso of the same patient, and applied our surface-fitting algorithm on them. The control points set n x m means that there is n control points in the u direction and m control pints in the v direction. The fitting results are listed out below:

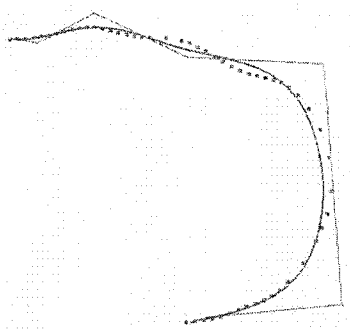


Figure 3.1: Curve approximation with 7 control points to 60 points

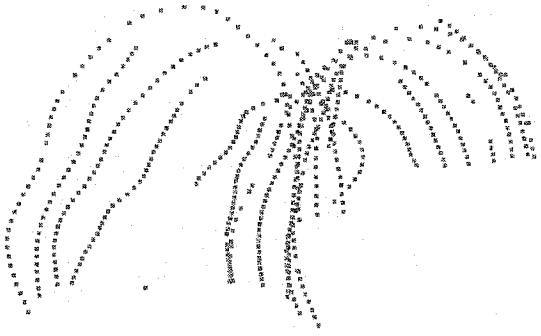


Figure 3.2: Raw data of half of the torso

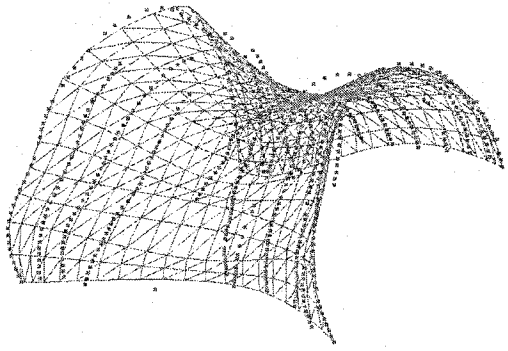


Figure 3.3: Surface approximation to figure 3.2 with a 10x4 control points set

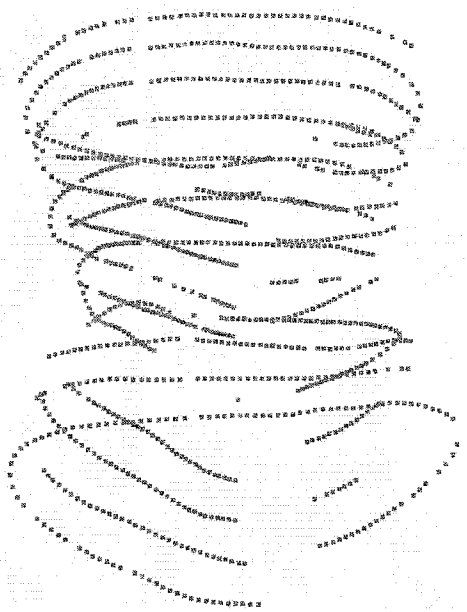


Figure 3.4: Raw data of a complete torso of 120x11, i.e., 11 contour lines and 120 points per line

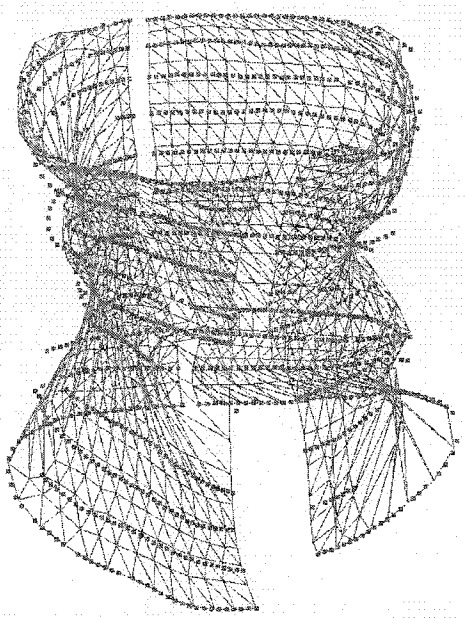


Figure 3.5: Surface approximation to figure 3.4 with an 8x8 control points set for each half torso.

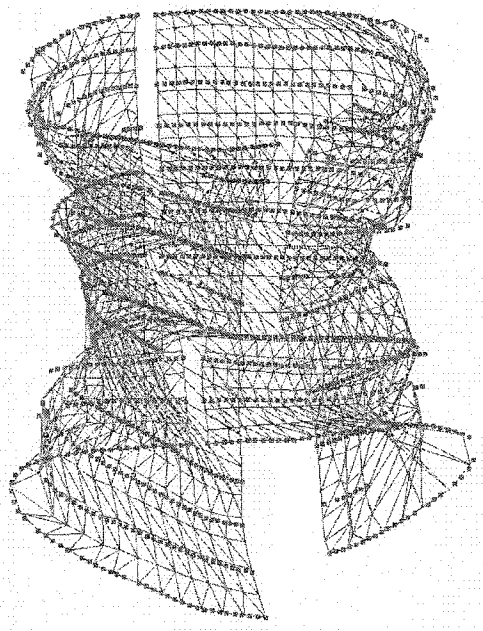


Figure 3.6: Surface approximation to figure 3.4 with an 11x8 control points set for each half torso.

3.2.2 Fitting Results on Raw Calgary Data

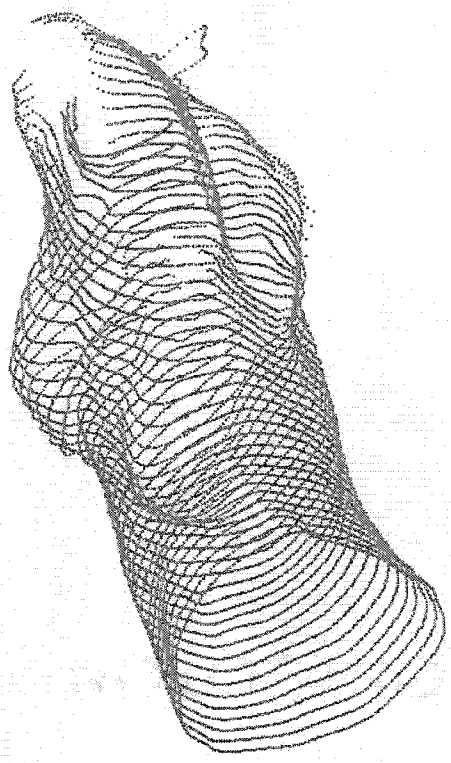


Figure 3.7: Raw data of a complete torso of 360x46, i.e., 46 contour lines and 360 points per line.

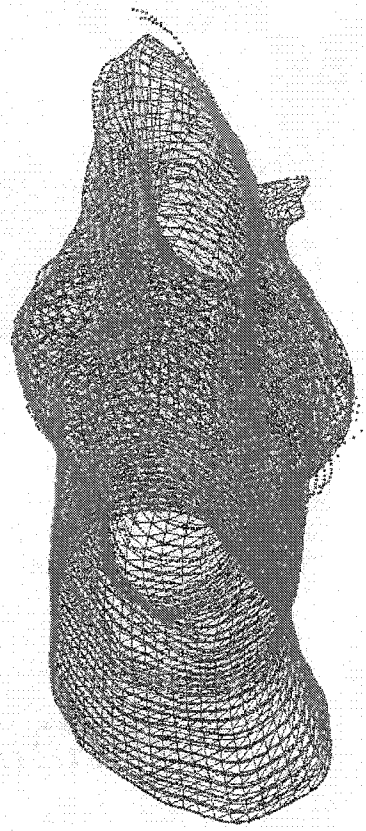


Figure 3.8: Surface approximation to figure 3.7 with a 41x11 control points set.

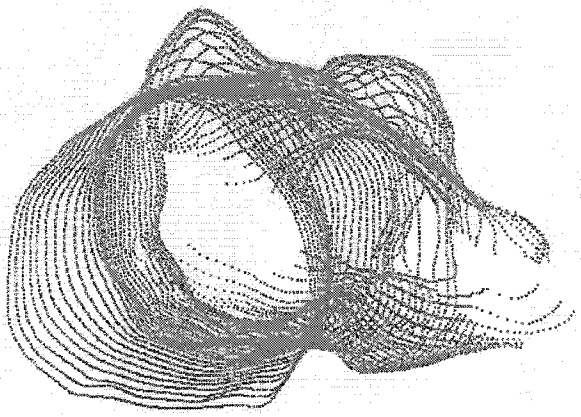


Figure 3.9: Same patient as in figure 3.7 from horizontal viewpoint.

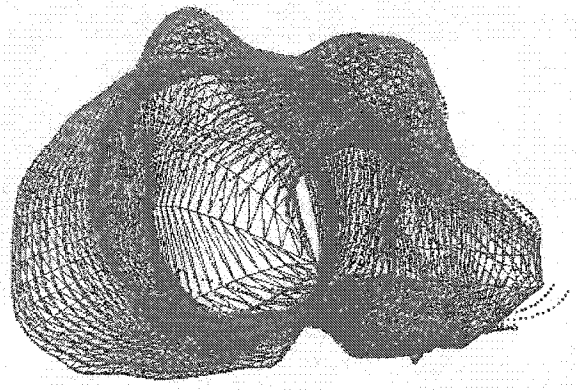


Figure 3.10: figure 3.8 from horizontal viewpoint.

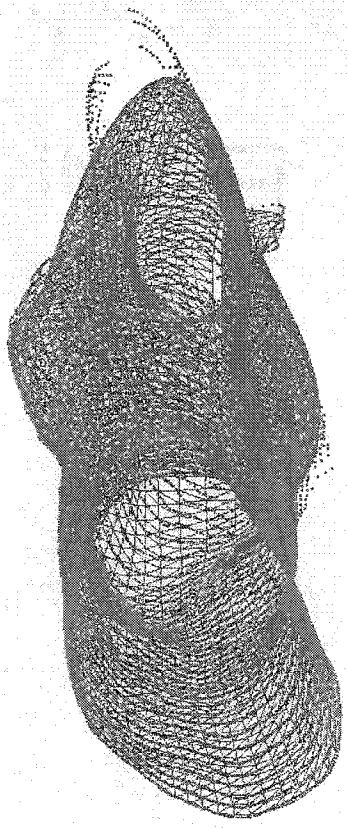


Figure 3.11: Surface approximation to figure 3.7 with a 16x8 control points set.

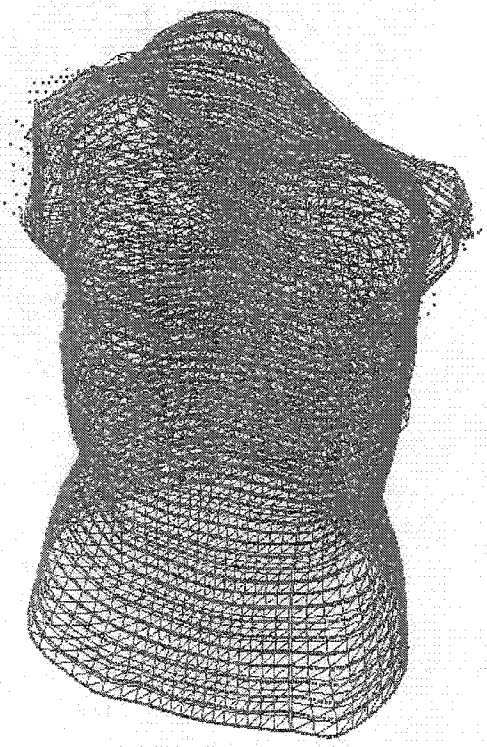


Figure 3.12: Surface approximation to figure 3.7 with a 61x31 control points set.

3.3 PCA Results

After fitting a surface to the two sets of raw points, we then applied the PCA technique to the obtained control points to reduce dimension by keeping 80% and 90% variance, respectively. For the data composed of control points set, we call it control point data; for the data processed by PCA technique, we call it PCA data. One thing we have to mention is that when applying PCA algorithm onto the datasets of 61x31 control point sets with 115 data, the algorithm failed to give results. By carefully checking, we found that it was because the memory requirement of the algorithm had exceeded 2G bits. We could not solve this problem because 32-bits windows program can maximally access no more than 2G bits memory. We tested a dataset of 61x31control point set with 89 data, the PCA algorithm worked. But after exceeded that limit, it always failed. Therefore, in the following results tables, the result on datasets of 61x31 control point set when applying PCA was obtained on the test dataset with 89 data, not on the full dataset with 115 data.

Data set	Control points net	Variance reservation ratio	Dimension of control point data	Dimension of PCA data
Brace data	16 x 8	80%	384	15
Brace data	16 x 8	90%	384	22
Calgary data	16 x 8	80%	384	4
Calgary data	16 x 8	90%	384	9
Calgary data	41 x 11	80%	1353	4
Calgary data	41 x 11	90%	1353	7
Calgary data	61 x 31	80%	5673	4
Calgary data	61 x 31	90%	5673	5

Table 3.3: PCA results

3.4 Parameter Tuning Results

On scoliosis dataset we committed parameter tuning experiments only on Calgary data with absolute clinical Cobb angle. We did not commit on parameter tuning on Brace data since it was only used for the purpose of the feasibility test of our method. We had three types of Calgary datasets. Each of them corresponded to different 'tightness' of fitting to the original raw data. There were datasets with 16x8, 31x11, and 61x31 control points, which represent loose, moderate and highly tight fitting, respectively. We give the optimal parameter values and their accuracy in the figures below.

- On 16x8 dataset, the optimal values found were: $C = 2^{11}$, gamma = 2^{-15}

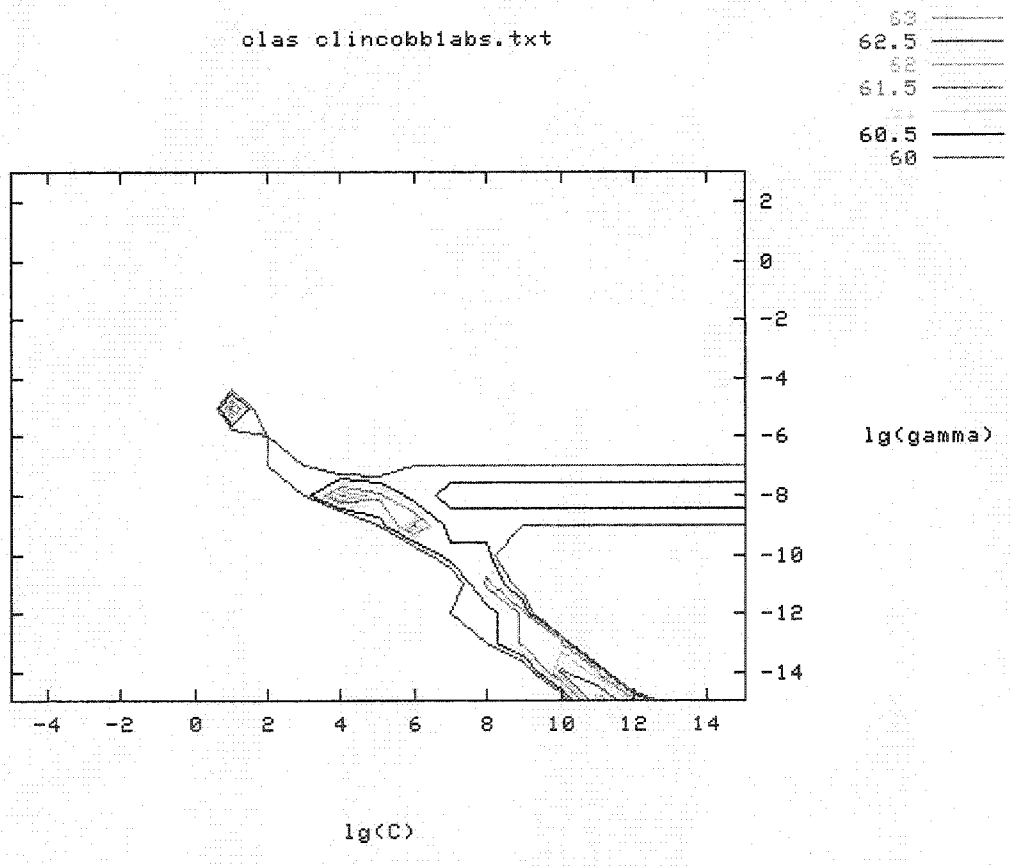


Figure 3.13: Performance accuracy with different parameters setting on 16x8

- On 41x11 dataset, the optimal values found were: $C = 2^{6}$, gamma = 2^{-11}

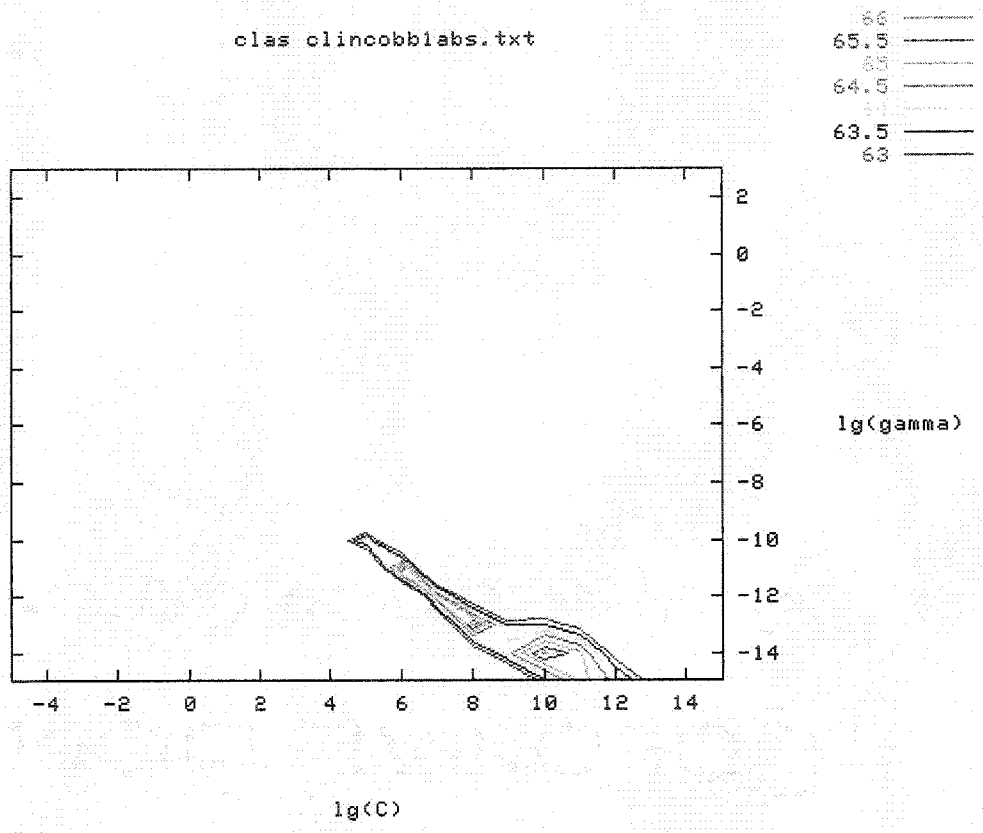


Figure 3.14: Performance accuracy with different parameters setting on 41x11

- On 61x31 dataset, the optimal values found were: $C = 2^{11}$, gamma = 2^{-15}

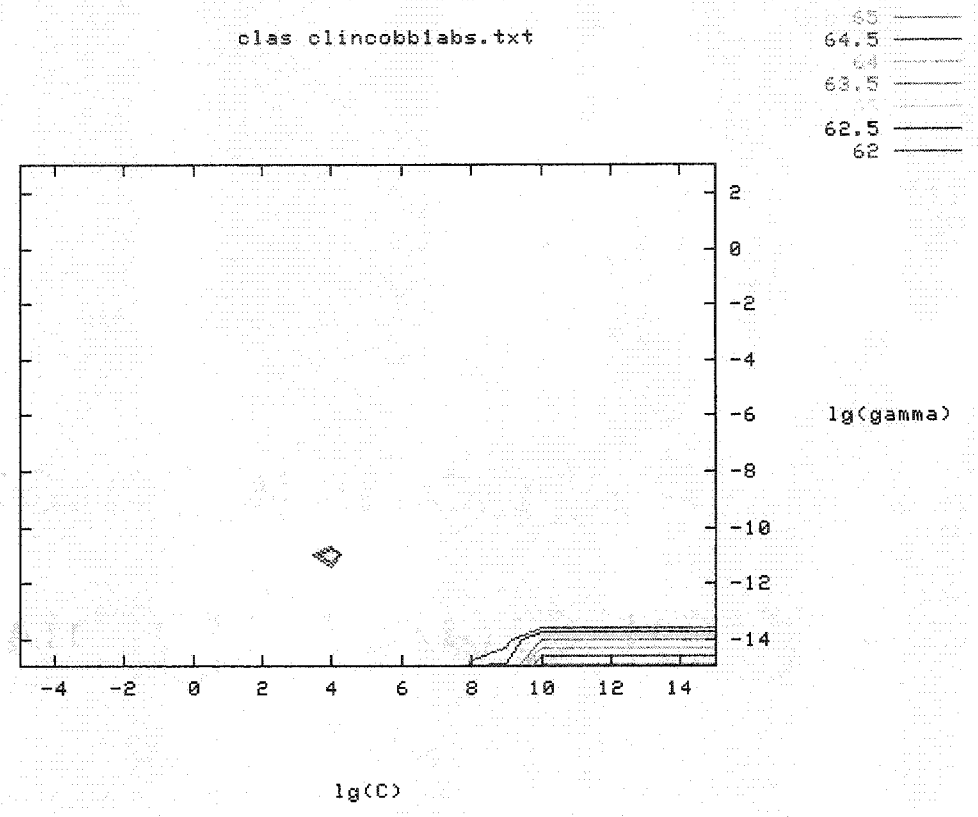


Figure 3.15: Performance accuracy with different parameters setting on 61x31

3.5 Classification Results

We committed classification experiments on two kinds of datasets: Brace dataset and Calgary dataset on both cases utilizing PCA and not utilizing PCA before sending data into SVM. When utilizing PCA, two cases were tested: 80% variation of the dataset was kept and 90% variation of the dataset was kept. Both training and testing criteria, cross validation and leave-one-out, were tested on Brace data. For Calgary data, since from the experience on Brace data leave-one-out worked better on small data set than 10-fold cross validation, we utilized only the leave-one-out criteria.

3.5.1 Brace Classification Results

For RBF kernel, we utilized the parameter tuning technique, and obtained that the optimal values were: C = 128, gamma = 1/2048. For all the others, we just randomly tested a number of values of the parameters and then chose the ones which performed the best. We obtained that C = 10, d = 3, gamma = 3. The percentage of support vectors among the training data (37 and 40, respectively) is also reported in the following tables.

- Keeping 80% variation when applying PCA

Leave-one-out:

Kernel	Training error	Test error	Support vectors
Linear	6.8%	43.9%	70.18%
Polynomial	0%	46.34%	75.67%
RBF	9.45%	48.78%	80.37%
ERBF	0%	34.15%	98.11%

Table 3.4: Brace classification result with 80% PCA and leave-one-out

10-fold cross validation:

Kernel	Training error	Test error	Support vectors
Linear	5.3%	50.5%	67.13%
Polynomial	0%	45.5%	75.32%
RBF	46.6%	78%	79.92%
ERBF	0%	36%	98.09%

Table 3.5: Brace classification result with 80% PCA and 10-fold cross validation

- Keeping 90% variation when applying PCA

Leave-one-out:

Kernel	Training error	Test error	Support vectors
Linear	3.12%	65.85%	58.72%
Polynomial	0%	39.02%	92.5%
RBF	39.39%	78.05%	80%
ERBF	0%	29.27%	100%

Table 3.6: Brace classification result with 90% PCA and leave-one-out

10-fold cross validation:

Kernel	Training error	Test error	Support vectors
Linear	1.9%	55.5%	60.4%
Polynomial	0%	40.5%	91.87%
RBF	46.61%	78%	79.65%
ERBF	0%	33.5%	99.45%

Table 3.7: Brace classification result with 90% PCA and 10-fold cross validation

- Without applying PCA

Leave-one-out:

Kernel	Training error	Test error	Support vectors
Linear	19.88%	53.66%	91.95%
Polynomial	0%	34.15%	98.11%
RBF	37.99%	75.61%	94.15%
ERBF	**0%**	**29.27%**	100%

Table 3.8: Brace classification result with no PCA and leave-one-out

10-fold cross validation:

Kernel	Training error	Test error	Support vectors
Linear	16.6%	58%	91.83%
Polynomial	0%	38.5%	98.37%
RBF	46.61%	78%	93.48%
ERBF	0%	35.5%	100%

Table 3.9: Brace classification result with no PCA and 10-fold cross validation

3.5.2 Calgary Classification Results

By applying the parameter tuning method, the optimal parameter set was found:

Datasets with 16x8 control point net: $C = 2^{11}$, gamma = 2^{-15}, epsilon = 0.001, degree = 3.

Datasets with 41x11 control point net: $C = 2^{6}$, gamma = 2^{-11}, epsilon = 0.001, degree = 3.

Datasets with 61x31 control point net: C = 2^{11}, gamma = 2^{-15}, epsilon = 0.001, degree= 3.
For each control point net setting, we got different datasets by using different Cobb angle to determine the class label value. The explanations are as follows:
Clincobb1: use the value of the primary clinical Cobb angle to determine the class label.
Clincobb1abs: use the absolute value of the primary clinical Cobb angle to determine the class label.
Clincobb1_positive: we took out those data with negative clinical Cobb angle and kept only the data with positive clinical Cobb angle in the dataset. This dataset was only used in the classification experiments, not in the regression experiments. The purpose of doing this was that we hoped to see better performance by eliminating the affection of the data with negative Cobb angle.
Mtlcobb1: use the value of the primary computer Cobb angle to determine the class label.
Mtlcobb1abs: use the absolute value of the primary computer Cobb angle to determine the class label.
Mtlcobb1_positive: same as in Clincobb1_positive.
A recent result by Keerthi and Lin (Keerthi and Lin, 2002) shows that if RBF is used with model selection, then there is no need to consider the linear kernel. In our case, the mechanism of scoliositic deformity transforming to surface deformities is very complex. Linear kernel is obviously too simple for this problem. Hence, in the following experiments, we didn't consider linear kernel any more. The percentage of support vectors among the training data (114) is also reported.

- Keeping 90% variation when applying PCA

Same as we discussed in the section of principal component analysis (2.3.3.), when applying PCA algorithm onto the datasets of 61x31 control point sets with 115 data, the algorithm failed to give results due to limitation of memory. Therefore, in the following results tables, the results on datasets of 61x31 control point set when applying PCA are not given.

Clincobb1_positive:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	29.66%	36.26%	67.41%
RBF 16x8	30.2%	40.66%	68.93%
Polynomial 41x11	29.26%	50.55%	69.71%
RBF 41x11	28.96%	51.65%	71.5%

Table 3.10: Calgary classification result with 90% PCA on Clincobb1_positive

Clincobb1abs:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	28.9%	36.52%	59.79%
RBF 16x8	29.8%	37.39%	60.94%
Polynomial 41x11	27.19%	42.61%	62.3%
RBF 41x11	29.05%	46.09%	63.49%

Table 3.11: Calgary classification result with 90% PCA on Clincobb1abs

Mtlcobb1_positive:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	34.1%	45.05%	44.41%
RBF 16x8	34.32%	46.15%	46.67%
Polynomial 41x11	26.67%	54.95%	47.72%
RBF 41x11	28.73%	54.95%	49.3%

Table 3.12: Calgary classification result with 90% PCA on Mtlcobb1_positive

Mtlcobb1abs:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	30.36%	45.22%	44.1%
RBF 16x8	32.15%	46.96%	44.74%
Polynomial 41x11	28.73%	50.43%	47.26%
RBF 41x11	28.99%	51.3%	48.33%

Table 3.13: Calgary classification result with 90% PCA on Mtlcobb1abs

- Without applying PCA

Clincobb1_positive:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	6.29%	34.07%	73.52%
RBF 16x8	**9.71%**	**32.97%**	73.89%
Polynomial 41x11	3.2%	47.25%	76.32%
RBF 41x11	5.69%	45.05%	76.57%
Polynomial 61x31	0%	42.86%	73.38%
RBF 61x31	0%	41.76%	72.82%

Table 3.14: Calgary classification result with no PCA on Clincobb1_positive

Clincobb1abs:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	10.83%	33.91%	64.28%
RBF 16x8	**10.78%**	**33.91%**	67.27%
Polynomial 41x11	3.38%	40%	70.22%
RBF 41x11	7.13%	36.52%	71.27%
Polynomial 61x31	0%	34.78%	65.93%
RBF 61x31	0%	34.78%	65.91%

Table 3.15: Calgary classification result with no PCA on Clincobb1abs

Mtlcobb1_positive:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	5.37%	40.66%	44.54%
RBF 16x8	8.75%	42.86%	46.42%
Polynomial 41x11	1.97%	40.66%	49.73%
RBF 41x11	4.77%	40.66%	49.83%
Polynomial 61x31	0%	36.26%	48.5%
RBF 61x31	0%	36.26%	48.46%

Table 3.16: Calgary classification result with no PCA on Mtlcobb1_positive

Mtlcobb1abs:

Kernel	Training error	Test error	Support vectors
Polynomial 16x8	9.47%	43.48%	46.54%
RBF 16x8	12.04%	42.61%	46.48%
Polynomial 41x11	3.49%	38.26%	47.28%
RBF 41x11	7.76%	36.52%	49.14%
Polynomial 61x31	0%	36.52%	48.01%
RBF 61x31	0%	36.52%	48.15%

Table 3.17: Calgary classification result with no PCA on Mtlcobb1abs

3.5.3 Prediction Simulation Results

In order to evaluate the performance of the SVM method we developed on the future coming new patients, we committed this preliminary prediction simulation experiment. The idea is as below: from the 115 scans of 48 patients at hand, we took out all the scans of three patients, each patient fell in one of the three classes respectively. The scans of these three patients composed of the prediction set (i.e., the test set), all scans of patients left composed of the training set. With this setting, the three patients never appeared in the training procedure, thus they could be viewed as the future coming new patients. In fact, we picked out patient 2213591, 4340957 and 9133211 (patient's ID). The 4 scans of patient 2213591 fell in class 2, the 5 scans of patient 4340957 fell in class 1, and the 2 scans of patient 9133211 fell in class 3. Thus, the size of test set was 11, and the size of training set was 104. Parameter tuning was carried out and the optimal values found for C and Gamma were C=1 and Gamma=1/64, respectively. The performance of the system is given in the following table:

Clincobb1abs:

Kernel	Training error	Test error
Linear 16x8	0%	61.54%
Polynomial 16x8	0%	57.69%
RBF 16x8	18.18%	51.92%

Table 3.18: Prediction result without PCA on Clincobb1abs

3.5.4 GA-SVM Classification Results

Our method does not involve any feature extraction process. The disadvantage of it is that its performance is usually worse than the performance of methods which use features. After hading finished the experiments with our method, we had a new idea. We wanted to replace ANN in the GA-ANN method by SVM in order to test how SVM would perform on Jaremko's data. Since this time SVM worked on the 17 features selected by GA, we called this new method GA-SVM. The only difference between GA-ANN and GA-SVM was that we used SVM instead of ANN to do the final classification task. In order to keep fairness, we set the experimental environment same as in the GA-ANN, i.e., the first four collections of 89 data were used as the training set; the fifth collection of 26 data was used as the test set; this test set was never involved in the training procedure; each data contained 17 features which were the same as used in GA-ANN; the data were divided into 3 classes according to their absolute clinical Cobb angles (<30, 30 ~ 50, and >50, respectively). These were exactly the same settings as in Jaremko's experiments. Parameter tuning was carried out on the training set. We obtained the following optimal parameter values and classification accuracy:

	C	Gamma
Linear	2^{-3}	
Polynomial	2^{-1}	2^{-14}
RBF	2^{3}	2^{-14}

Table 3.19: Parameter tuning results of GA-SVM

Kernels	GA-SVM		GA-ANN	
	Training Accuracy	Test accuracy	Training accuracy	Test accuracy
Linear	95.51% (85/89)	80.77% (21/26)	93% (83/89)	92% (23/26)
Polynomial	94.38% (84/89)	80.77% (21/26)		
RBF	88.76% (79/89)	80.77% (21/26)		

Table 3.20: Classification results of GA-SVM with comparison to that of GA-ANN

3.6 Results Analysis

In learning tasks, what people care about more is the generalization performance, i.e., the test performance other than the training performance, because a learner who can perform excellently on the training data, but poorly on the future-coming data (i.e., test data), is almost useless. This kind of overfitting effect is what we always want to avoid. So, in the evaluation of the performance of each SVM learner on our two datasets: Brace dataset and Calgary, we compare them based on two principles: accuracy on test set and less overfitting.

3.6.1 Brace Results Analysis

The SVM which obtained the lowest test error was the one with ERBF kernel. The same results were obtained in both cases of applying PCA by keeping 90% variation and without applying PCA; the training method was leave-one-out. Under this setting, the training error was 0, and the test error was 29.27% (Table 3.6, Table 3.8).

As the pilot study of experimenting on Calgary data, we tested all of our methods (including surface fitting, PCA, SVM classification) on these Brace data. The test error rate was high on the Brace dataset. The reasons were various. The first source of the high test error rate was the overfitting effect. SVM performed much better on the training set than on the test set, e.g., in most classification experiments with any kernel, the training errors were commonly zero, nonetheless the test errors were much higher than zero. This is obviously an overfitting effect. The possible reason is that we encountered the well known phenomenon of "the curse of dimensionality", i.e., when the dimensionality of the inputs is high, the numbers of data that are needed to guarantee good results are huge, unless we impose some restrictions on the class of function to be approximated. Since the dimension of our data was very high (384 in our experiment) nonetheless we had only 41 data for both training and testing, it is not surprising that the results were overfitted. The

overfitting effect is common with small sample sizes and would likely be reduced as more patients' data are collected.

The second source of the high test error rate was the labeling of the data. As we explained before, the labeling was done by looking at the Cobb angle of each patient. But the problem is that one patient may have up to 5 Cobb angles from the PA view. What we did was simply picking one of them (usually was the Cobb_PA_2) and comparing it to the threshold to decide that patient's class. Obviously this method was rough and imprecise. By carefully checking those data, we found that there were so much diversity among the data of same class, namely, dividing these data into more classes is undoubtedly necessary.

The third source of the high test error rate was the chaotic status of the original scattered points. When these 3D digitized points were collected, many of them were missing or misplaced (we can clearly see this phenomenon from the surface fitting figures in section 2.2), which directly led to the inaccuracy of the surface fitted to these points. In a word, the original data was full of heavy noise. No classifier could perform well under this kind of situation.

The last source of the high test error rate, which was also the most fundamental one, was that the patients in the Brace datasets were all braced, which means their torso surface were drastically '*corrected*', while the internal scoliotic spine were not. Bracing mechanically altered the normal relation between torso surface and spinal deformities (Labelle et al., 1996), we risked degrading our correlation.

3.6.2 Calgary Results Analysis

The SVM which obtained the lowest test error was the one with RBF kernel on the 16x8 Clincobb1_positive dataset without applying PCA. The training error was 9.71% and the test error was 32.97% (Table 3.14). The second best result was still obtained by RBF

kernel on the 16x8 Clincobb1abs dataset without applying PCA either. The training error was 10.78% and the test error was 33.91% (Table 3.15), which were very close to the best results.

Similarly to the Brace data case, the test error rate was high on Calgary data too. Overfitting effect also occurred here. One possible reason was still the same as in the Brace data case, i.e., the small sample size leaded to overfitting. Although the size of Calgary dataset (115 data) was larger than the size of Brace dataset (41 data), it was still too small for efficient training and testing at the presence of high dimensionality of each data (384, 1353, and 5673, respectively). Especially in the classification experiment of Calgary data, we had three classes other than two classes in the Brace data case. At the part of data characteristics we had seen that the data in these three classes were highly unbalanced (see 2.5.2.2). There were 59 data in class one; however, there were only 11 data in class three. 11 data was obviously not enough for sufficient training and testing. The high percentage of support vectors also demonstrated the fact that the size of available data was far from being sufficient for training. Typically speaking, if the SVM captures the distribution of the data well, the percentage of support vectors should be relatively low.

According to the description of the characteristics of the Calgary data in (Jaremko, 2001), among the 115 Calgary data:

- 63 were from braced patients, the other 52 were not.

As we know, bracing mechanically altered the normal relation between torso surface and spinal deformities (Labelle et al., 1996), we risked degrading our correlation. So these 63 data in fact played a role of '*noise*' in the learning procedure.

- 91 patients had rightward curve, 24 patients had leftward curve.

When committing experiments with the absolute Cobb angle to determine the class label value of each patient, all the 24 patients with leftward curve were treated as having same-magnitude rightward curve. So these 24 data in fact played a role of '*noise*' in the

learning procedure, because the SVM was trying to learn that the leftward curve was the same as the rightward curve.

- 60 patients were under 13 years of age, 55 patients were above 13 years of age.

Most scoliotic patients in our dataset were girls (same thing in general). Generally speaking, for the girls above 13, their breasts made their frontal torso surface very different from the girls below 13. Nonetheless, we took the 360^{o} full torso into account. Thus, the surface fitted to the girls above 13 would rather different from the surface fitted to the girls below 13. This phenomenon also played a role of '*noise*' in the learning procedure.

- Correlations among data

The data set consisted of multiple scans of the same patients (115/48 = an average of 2.4 scans per patient). While scans were separated by intervals of at least 6 months and involved differences in patient growth and posture, they were still not entirely independent. This kind of correlation would also cause problems.

In the raw data points of Calgary data, shoulder part of each patient was cut, which introduced problems into our surface fitting procedure. We can clearly see the effect of cutting from the figures of surface fitting (see 2.2.6.3). The cutting leaded to non-continuity at the upper part of each patient's data, the fitted surface could not correctly and precisely pass through this part. What we did was to let the fitted surface smoothly approximate these discontinuous points in the least mean square sense. This made us have almost the same shape at the shoulder part for all the patients. But in fact patients with different severity of deformed spine have drastically different shape on the surface at the shoulder part. So the missing points around should part also played a role of '*noise*' in the learning procedure.

Another factor which affected the accuracy of our experiments was the reliability of the Cobb angle. According to past research, an important problem with the Cobb angle is inter- and intra-observer variability in its measurement. For example, the standard deviation of inter-observer variability between four readers of 30 X-rays of scoliosis

patients was 2.5° (Goldberg et al., 1988), for a 95% confidence interval of nearly $\pm 5^0$ (assuming a symmetric normal distribution), while Cobb angle measurements on 50 X-rays differed by 7.2° between four orthopaedic surgeons when they were allowed to select curve endpoints individually. These variations in Cobb angle measurement made it inaccurately reflect the class label of each patient. For those patients whose Cobb angle were around the class thresholds (i.e., 30° and 50°), they could have been fallen into another neighbouring class if the variation of its Cobb angle was taken into account. This phenomenon also played a role of '*noise*' in the learning procedure.

Another main reason was probably our method. Because Jaremko worked on the same raw data set, this means that he suffered from the same difficulties as we mentioned above too. However, his GA-ANN method obtained excellent results from it. Since from the GA-SVM experiment we have seen that SVM can acquire almost the same performance as ANN on the same data set, the only possible explanation to why our method performed worse than the GA-ANN method should be that the methods making use of features suffered from those 'noises' existing in our data sets much less than the methods which do not make use of features at all (e.g., our method). We also realized afterward that the control points of a surface model were probably not the best representation of the surface deformities. We will come back to this subject in more details in the discussion section.

3.6.3 Prediction Simulation Result Analysis

The result of the prediction simulation experiment was consistent with the results we had in other experiments, i.e., polynomial kernel performed better but was overfitted, RBF kernel performed less accurately than polynomial but was not overfitted. Since we simply randomly chose three patients as the test data, the results obtained on these three patients could not represent the general performance of the SVM method we developed. Choosing different patients can lead to different training and testing accuracy. This was just a preliminary test of our system. The kernel point of this experiment was that we did not

want to see the same patient appear in both training and testing sets. This is different from the performance evaluation experiment Jaremko committed. In his experiment, he utilized the fifth collection of data as the test set, and the data of the test set was never used in the training procedure. However, the problem with his approach was that most patients in the test set also appeared in the training set, i.e., these patients took several scans at different data collection periods. By carefully checking the original data, we found that the Cobb angle of most patients didn't significantly changed among different scans. This means that the scans from different period were in fact almost the same. Thus, one potential problem that Jaremko's evaluation method might bring is that the same data used in the training will be used again in the test procedure. This is obviously an affect we don't want to have. The purpose of our prediction simulation experiment was to try to check the performance of the system by avoiding this problem.

3.6.4 GA-SVM Results Analysis

The training accuracy of the GA-SVM is almost the same as that of the GA-ANN, but the test accuracy of the GA-SVM is slightly worse than that of the GA-ANN. SVM correctly classified 21 out of 26 data, only 2 data less comparing to ANN's performance on the same data set. This is reasonable. Because according to (Jaremko, 2001), the GA-ANN performance reported was the result of the best run of many runs, and the same ANN which was used to classify the test data was used as the fitness function of the GA, which means that this ANN was optimized for this data set. Therefore, it is normal that slightly better result was obtained on the test set by ANN other than by SVM. But generally speaking we may say that these two results are almost the same. The significance of the GA-SVM experiment is that it proved that the SVM method we utilized in our approach is as powerful as the ANN Jaremko utilized in his GA-ANN approach; it also indirectly showed the control points did not capture the surface deformities as efficiently as those 17 features, which implied other better representation should be sought for.

3.7 Discussion

From the experiments and results on the above experiments, we had following findings and discussions:

- Under the condition of small sample set, leave-one-out training method can usually lead to better results than 10-fold cross validation.

This is consistent with the theoretical analysis we discussed before (section 2.4.7). From the experimental results we obtained on Brace data by using both training methods, we also found that leave-one-out worked better than 10-fold cross validation. Basing on the reality that it is usually very hard to collect plenty of scoliotic patients in short term, we think leave-one-out is more appropriate to be used in our problem. That is why we utilized only leave-one-out other than 10-fold cross validation method on the Calgary data. However, there is a price to pay for this benefit. Leave-one-out is much more time-consuming than 10-fold cross validation, since its iteration times is equal to the number of the total data. As SVM is a type of very fast learning machine, the degree of time consumed on the data set of scale of hundreds of data points is still acceptable.

- By keeping more variation when applying PCA dimension reduction technique, better results can be obtained than keeping less variation.

This is also consistent with the theoretical analysis, because keeping more variation means more information is reserved. This is certainly beneficial. Our results also supported this conclusion. However, we also have to point out that PCA does not always help in improving the performance. The advantage of PCA is that it helps to reduce the dimension of the data. Handling low-dimension data is usually easier and faster. The disadvantage is that some information of the original data will be lost during the procedure of dimension reduction. This can sometimes lead to the decline of the performance. Both phenomenons were observed in the experiments on the Brace data. From the experience with the Brace data, we still tested both settings of applying PCA and without applying PCA on the Calgary data. But when applying PCA, only the setting of keeping 90% variation was adopted, since from the experiences on Brace data we learned that it is unnecessary to try by keeping 80% variation any longer. From the

experimental results on the Calgary data we observed that both the best and the second best results were obtained by without using PCA technique. This is reasonable. Because at the surface fitting step, we had already lost a part of information of the original raw data (this is simply because fitted surface can't 100 percent pass through all of those data points), by using PCA more information were lost. The advantage of PCA was also observed in the experiments with the Calgary data, namely, PCA significantly reduced the time required by the learning procedure, especially on the extremely-high-dimension data. For instance, on the 61x31 data sets, SVM needed about 10 minutes to finish the learning and testing without using PCA. By using PCA, since the dimensionality of data was drastically reduced that SVM needed only a few seconds to finish the learning and testing. In clinic, the above-all requirement is the accuracy of the method. Therefore, we can say PCA didn't work successfully in our case and we suggest not using it in the future study.

- Increasing the density of the control points did not significantly improve the classification performance in out method. Both the best and the second best performance were obtained on 16x8 data sets. At the first look, this conclusion seemed irrational, since increasing the density of the control points should consequently increase the quality of the surface fitted to the raw scanned data, and hence more precise representation (the fitted surface) should lead to better classification performance. In fact, from the view point of the quality of the fitting surface, increasing the density of the control points will surely lead to better result; but from the view point of learning, that is not certain. Because the problem that increasing the density of the control points introduced was that it increased the dimension of the dataset. For example, on the Calgary data, we tested three sets of density of control points; they were 16x8, 41x11, and 61x31, respectively. The quality of the fitted surfaces with these three sets of control points were given in section 2.2.6.3. The quality of the surface increased with the increasing of the number of control points. But the dimension of dataset varied from 384, 1353, to 5673. At the presence of having only 115 data for both the training and the testing, these dimensionalities were really too high. Therefore, it was not surprising that increasing the

density of the control points did not consequently increase the classification and regression performance. Another severe problem caused by high-density control points set was the distortion at the non-continuous part. For example, when the number of control points was increased to 61x31, the surface at the shoulder part was distorted (see figure 3.12), this was because the arm removal operation eliminated many points at the shoulder part. On smoothly continuous data this phenomenon should vanish.

However, we must point out that the fundamental problem was not originally caused by increasing the density of the control points. As we discussed before, the fundamental cause was from using control points as the representation of the surface deformities. The control point is not a sort of steady representation, it is variable and very sensitive to any slight changes occurred on its corresponding surface model. For instance, the same patient takes surface scans at two different times (e.g., an interval of half a year), the shape of patient's trunk might have some small changes such as the patient becomes slightly fatter, or thinner, or taller, these changes highly possibly happen since most of scoliotic patients are teenagers. Then the spatial location of the control points of the patient's surface models can vary significantly due to the change on the surface model. But the deformities of the inner spines in most cases stay rather steadily in this relative short period. In the extreme case, even the same patient takes two scans at the same day could have different control points set simply because the patient did not stand with the same posture during the two scanning. This is a serious problem of using the control points as the representation of the surface deformities. Hence, more steady and invariant representations of the surface deformities should be sought for in a future study.

- Performance on all the datasets: Clincobb1, Clincobb1abs, Clincobb1_positive, Mtlcobb1, Mtlcobb1abs, Mtlcobb1_positive, were similar.

This is consistent with Jaremko's results. The correlation between clinical Cobb angle and computer Cobb angle was very high (Jaremko, 2001), so setting class label with clinical Cobb angle or computer Cobb angle did not matter much. As for Clincobb1abs and Mtlcobb1abs, i.e., using the absolute value of the Cobb angle to determine the class label, because the ratio of negative Cobb angle was rather low in the whole dataset (for

instance, in Clincobb1, there were only 24 data with negative Cobb angle, the ratio was only 24/115 = 21%), they could not significantly affect the result either.

- On Calgary data, the performance of polynomial kernel and RBF kernel were very close, although RBF kernel performed generally slightly better than polynomial kernel. This is consistent with SVM theory, namely, different choice of kernels in applying SVM to classification problem should not cause significant change in the performance. In the case of without using PCA, both polynomial and RBF kernels obtained zero training error on the 61x31 data sets, this was due to the extremely high dimensionality of the data sets. So the decision hyperplane found on these training sets was overfitted and this leaded to the high test error. Thus, basing on the experimental results on the Calgary data, we suggest using RBF or ERBF kernel as the choice of kernel function of SVM for the future study. It could obtain good performance and less overfitting at the same time. When more patients' data are collected, the performance of SVM can be expected to be improved.

In recent years, researchers in machine learning domain found that the ensemble of a group of learners often outperformed than the single base learner. This kind of new methodology is call ensemble learning. Based on the fact that our SVM did not perform well enough in our experiments, we would like to know whether the ensemble of SVMs could work better or not. So, we did an investigative study in constructing an ensemble of SVMs, and tested its performance on some artificial data. The test results were quite promising. We did not test the ensemble of SVMs on our scoliosis dataset, because the ensemble of SVMs is usually designed for large-scale learning problem, our scoliosis datasets (Brace dataset and Calgary dataset) are not suitable for it. When more data are collected, we can utilize this new methodology. We presented the details of constructing an ensemble of SVMs in the next chapter. This is an independent chapter and can be read separately from other chapters.

CHAPTER 4 - INVESTIGATIVE STUDY: THE ENSEMBLE OF SVMs

The ultimate goal of designing pattern recognition systems is to achieve the best possible classification performance for the task at hand. Support vector machines and ensemble methods, e.g. boosting, are two major new research directions toward this goal. As a novel and promising learning technique, SVM has shown its superiority to most traditional learning methods, such as decision tree, nearest neighbor etc., in diverse applications. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. The idea is not to rely on a single decision making scheme. Instead, all the designs, or their subsets, are used for decision making by combining their individual opinions to derive a consensus decision. Recent research has revealed from both theoretical and empirical aspects that ensembles can, more often than not, increase predictive performance over a single model (Dietterich, 2000; Schapire et al., 1998; Breiman, 1996). Various classifier combination schemes have been devised, such as bagging, boosting, stacking, error-correcting output codes etc. Generally people construct the ensemble over relatively *weak* learners, such as decision stump and C4.5. The ensemble of *strong* learners has been studied too. For instance, the combination of ensembles of neural networks (based on different initializations) has been studied in the neural network literature (Cho and Kin, 1995; Hansen and Salamon, 1990; Hashem and Schmeiser, 1995; Krogh and Vedelsby, 1995; Rogova, 1994). There are two severe problems existing in the ensemble methods carried out so far:

1. They don't perform well on noisy data
2. Sometimes they lead to over fitting.

As we know, SVM is a strong learner, it has excellent performance on all kinds of datasets compared to other learners, including on the above two 'bad' cases, i.e. it is robust to noise and it less likely leads to over fitting. So naturally we are wondering if it is possible to integrate the strengths of ensemble methods and SVM together, i.e., if we

can construct an ensemble by using SVM as the base. In fact, the idea of an SVM mixture is not new. Previous attempts include Kwok's work on Support Vector Mixtures for classification and regression problems, in which he did not train the SVMs on part of the dataset but on the whole dataset and hence could not overcome the time complexity for large datasets (Kwok, 1998). In order to overpass this limitation existing in Kwok's method, namely, to extend the ability of the ensemble of SVMs can handle very large scale problems, e.g., a dataset of hundreds of thousands examples, a group of researchers proposed a new mixture of SVMs very recently that can be easily implemented in parallel and where each SVM is trained on a small subset of the whole dataset (Collobert et al., 2002). Fernandez's work also revealed that training many local SVMs instead of a single global one can lead to significant improvement in the performance of a learning machine in a time series prediction application, as shown in (Fernandez, 1999).

An interesting issue in the research concerning classifier ensembles is the way they are combined. What is the best approach to constructing an ensemble of classifiers? In principle, there is no single best ensemble method, just as there is no single best learning algorithm. However, some methods may be uniformly better than others. Although some ways of constructing the ensemble of SVMs already exist, we noticed that the widely used combining method, AdaBoost, which is simple but very efficient, has not been tested yet in the assembling of SVMs. As a preliminary and investigative test, we tried to use construct the ensemble of SVMs with the standard AdaBoost method. Due to the past big success of ensemble methods, it can be expected that the ensemble of SVMs could outperform than a single SVM, and could overcome the above two problems. Motivated by this hypothesis, we investigated the applicability of the ensemble of SVMs with AdaBoost.

4.1 AdaBoost

In this part, we briefly review the algorithms and characteristics of AdaBoost. The algorithm we give out here is for the two-class case.

Given: m examples $(x_1, y_1),...,(x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1,+1\}$

Initialize $D_1(i) = 1/m$ for all $i = 1...m$

For t = 1,...,T:

- Train base classifier using distribution D_t.
- Get a hypothesis $h_t : X \to \{-1,+1\}$ with error

$$\varepsilon_t = \Pr_{i\sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i)\neq y_i} D_t(i).$$

- Choose $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$
$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)).$$

Figure 4.1: Pseudocode of AdaBoost

The main idea of this algorithm is to maintain a distribution or set of weights over the training set. Initially, all weights are set equally, but in each iteration the weights of incorrectly classified examples are increased so that the base classifier is forced to focus on the '*hard*' examples in the training set. For those correctly classified examples, their

weights are decreased so that they are less important in next iteration. Note that the error is the sum of weights of the misclassified instances divided by the total weights of all instances, instead of the fraction of instances that are misclassified.

The basic intuition why ensembles can improve performance is that uncorrelated errors made by the individual classifiers can be removed by voting. Another cause is that our hypothesis space H may not contain the true function f. Instead, H may include several equally good approximations to f. By taking weighted combinations of these approximations, we may be able to represent classifiers that lie outside of H. The following figure illustrates this idea:

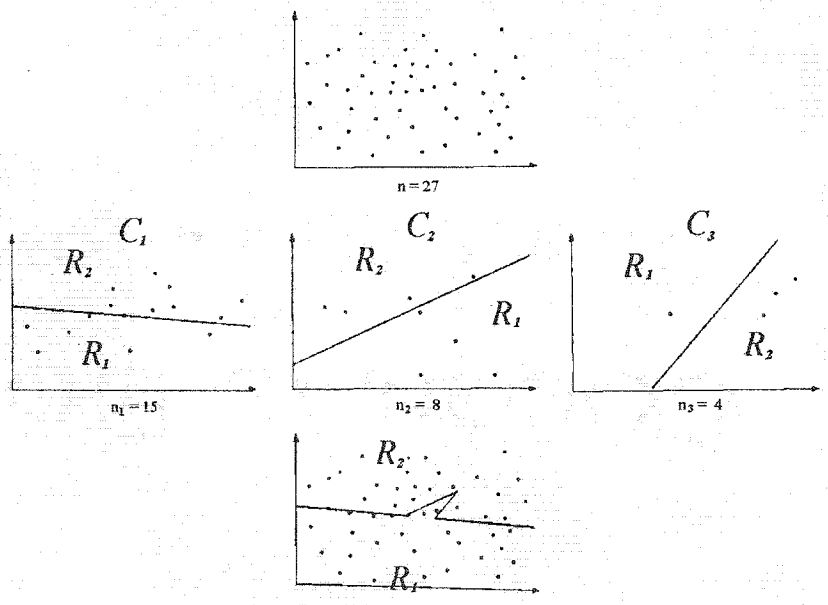


Figure 4.2: Illustration of boosting

In this example, a two-dimensional two-category classification task is shown at the top. The middle row shows three component (linear) classifiers trained by LMS algorithm, where their training patterns were chosen through the basic boosting procedure. The final classification is given by the voting of the three component classifiers and yields a nonlinear decision boundary, as shown at the bottom.

Because AdaBoost focuses on difficult training patterns, the training error of each successive component classifier (measured on its own weighted training set) is generally larger than that of any previous component classifier. Nevertheless, so long as the component classifiers perform better than chance (e.g. have error less than 0.5 on a two-category problem), the weighted ensemble decision ensures that the training error will decrease. It is often found that the test error decreases in boosted system as well, as shown in the following figure:

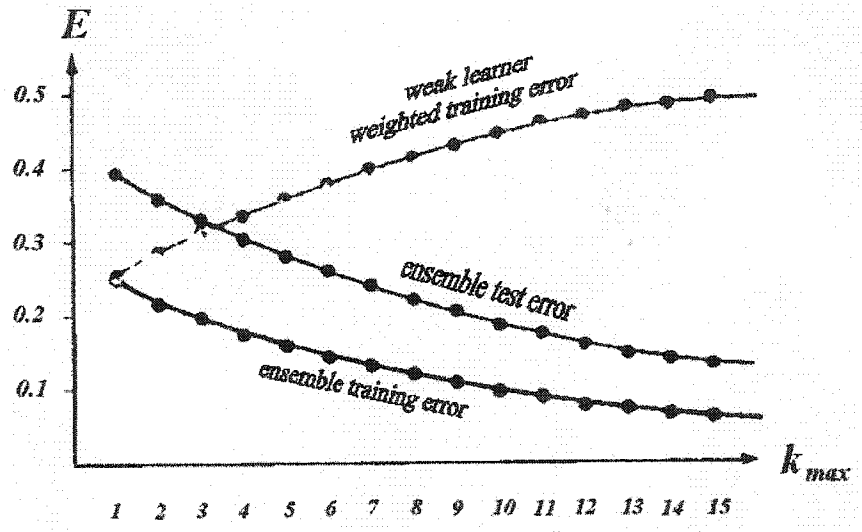


Figure 4.3: Characteristics of boosting

4.2 Constructing the Ensemble of SVMs

Now the question is: How to construct the ensemble of SVMs? We use a greedy optimization procedure to construct the ensemble of SVMs. We start with the case where weak features are linear decision rules

$$\phi_k(x) = sign\{(x \cdot w_k) + b_k\}$$

Our goal is to find N optimal hyperplanes that in greedy fashion minimize the functional

$$R(w,b) = \sum_{i=1}^{l} \exp\{-y_i \sum_{k=1}^{N} d_k sign[(x_i \cdot w_k) + b_k]\}$$

and then using these linear decision rules as the features to construct the desired ensemble.

4.2.1 Constructing the Features

To construct N features we need to specify in the general scheme described in the previous section only the method for minimizing the functional $R(\phi) = -\sum_{i=1}^{l} c_i^{k+1} y_i \phi(x_i)$ in the set of linear decision functions:

$$\phi_k(x) = sign\{(x \cdot w_k) + b_k\}$$

(Defined by the optimal hyperplane).

As before, we replace this problem with the following problem: Minimize the functional

$$R(w_k) = \frac{1}{2}(w_k \cdot w_k) + C\sum_{i=1}^{l} c_i^k \zeta_i^k, \quad c_i^1 = 1 \quad (1)$$

subject to constraints

$$y_i((w_k \cdot x_i) + b_k) \geq 1 - \zeta_i^k, \quad \zeta_i^k \geq 0$$

The only difference in the problem of constructing this hyperplane compared to the problem of constructing the soft-margin hyperplane described before is that in the case of the soft-margin hyperplane all coefficients c_i^k were equal to 1. Now the second term in (1) is a weighted sum.

We solve this optimization problem using the same technique with Lagrange multipliers. We obtain the following solution:

$$w_k = \sum_{i=1}^{l} y_i \alpha_i^k x_i$$

where the coefficients α_i^k maximize the functional

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (2)$$

subject to the constraints

$$0 \le \alpha_i \le Cc_i^k$$

and the constraint

$$\sum_{i=1}^{l} y_i \alpha_i c_i^k = 0$$

The coefficient b_k can be defined from Kuhn-Tucker conditions

$$\alpha_i (y_i (w_k \cdot x_i + b_k) - 1 + \zeta_i^k) = 0$$

Therefore, the difference in decision rules is defined by the coefficients c_i^k. these coefficients are calculated iteratively in a greedy optimization procedure:

$$c_i^1 = 1, \quad i = 1,....,l$$

$$c_i^{k+1} = \exp\{-y_i \sum_{r=1}^{k} d_r \phi_r(x_i)\} = c_i^k \exp\{-y_i d_k \phi_k(x_i)\},$$

where

$$d_k = \frac{1}{2} \ln \frac{\sum_{\{i: y_i \phi_k(x_i)=1\}} c_i^k}{\sum_{\{i: y_i \phi_k(x_i)=-1\}} c_i^k}$$

Differences: The coefficients c_i^k also play a role of the weight of the data here, which is the same as in AdaBoost. It is still used in the constrain condition for solving the dual problem, i.e. it affects the construction of decision boundary. So it has been used twice in each iteration. This is the different point between the ensemble of SVMs and the AdaBoost.

Analysis of the usefulness of c_i^k : in SVM, the constant C plays a role of the regularization parameter; a larger C corresponds to assigning higher penalty to errors. Now by multiplied by c_i^k, Cc_i^k can 'adaptively' vary with the error rate of the classifiers, and since Cc_i^k is used in the constrain conditions for solving equation (2), it will surely affect the solution of the Lagrange multipliers, and eventually affect the construction of SVM. The higher error rate a SVM make, the higher weight it will be assigned. These weights 'force' the SVM of next iteration to focus on those 'difficult' examples. It's exactly the same as in AdaBoost.

4.2.2 Constructing the Decision Rule

To obtain the decision rule one constructs the optimal hyperplane in *N*-dimensional binary space

$$z = (\phi_1(x),....,\phi_N(x)).$$

Using the given set of training data one obtains the new set of training data

$$(y_1, z_1),....,(y_l, z_l)$$

$(z_i = (\phi_1(x_i),...,\phi_N(x_i))$, based on which one constructs the optimal hyperplane. So the final decision rule is

$$H(x) = sign\left(\sum_{d=1}^{N} d_k \phi_k(x)\right)$$

It is simply a weighted voting as in AdaBoost.

4.2.3 Ensemble of Nonlinear SVMs

In the case where decision function is not a linear function of the data, we need to employ a so-call "*kernel function*" *K*. What this kernel function does is a mapping of the data from original space to a (possibly infinite dimensional) feature space *H,* i.e. Hilbert space. In Hilbert space, data are linearly separable, so we still construct a linear SVM in Hilbert space to separate mapped data. This linear SVM in Hilbert space is corresponding to a nonlinear SVM in the original space.

We can use features of the form

$$\phi_k(x) = sign\left(\sum_{i=1}^{l} y_i \alpha_i K(x, x_i)\right)$$

where the Lagrange coefficients α_i are solution of the following optimization problem:

Maximize the functional

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j)$$

subject to the constraints

$$0 \le \alpha_i \le C c_i^k$$

$$\sum_{i=1}^{l} y_i \alpha_i c_i^k = 0$$

We can see that this is almost the same as in the linear SVM case. The only difference is the kernel function. In fact, linear SVM is just a special case of the nonlinear SVM, where the kernel function is a polynomial kernel and degree is one.

Using obtained N features $\phi_k(x)$, $k = 1,...,N$, we can construct the final decision rule, which is simply a weighted voting as in AdaBoost, i.e.

$$H(x) = sign\left(\sum_{d=1}^{N} d_k \phi_k(x)\right)$$

We can see that in the ensemble of nonlinear SVMs case, every step is the same as in the construction of the ensemble of linear SVMs case, except the use of a kernel function.

4.3 A Further Improvement

In the above-discussed method for constructing the ensemble of SVMs, we use the standard AdaBoost algorithm to reweigh training data at each iteration and produce the final decision rule with a weighted majority voting. Recently the concept of margin was drawn into the research of the efficiency of boosting, and it shows that the margin distribution of AdaBoost resembles the one of SVMs for the separable case. In fact, AdaBoost achieves a *hard margin* asymptotically, such as the SVMs for the separable case. But at the presence of classification noise, the decision boundary created by a hard margin classifier will become too complex and *'sticked'* on the training data (in order to correctly classify all training data), i.e. it overfits!

The following figure illustrates this problem:

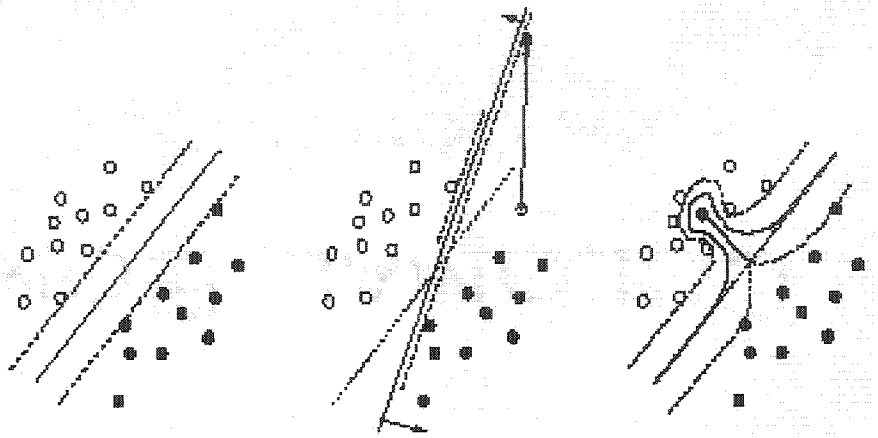


Figure 4.4: Hard margin

The three graphs show the maximum margin hyperplane found one three dataset, respectively: on reliable data (left), on data with an outlier (middle) and on data with a mislabeled pattern (right). This figure shows that the hard margin implies noise sensitivity, only one pattern can spoil the whole estimation of the decision boundary. In fact, most observed poor performance of AdaBoost was also found on noisy datasets.

A variety of AdaBoost, which is call regularized AdaBoost, suggested a nice solution to this problem. It analogously used the idea of *soft margin* as in SVM research. Firstly we define the margin for an input-output pair $z_i = (x_i, y_i), i = 1,...,m)$ by

$$mg(z_i, \mathbf{c}) = y_i \sum_{t=1}^{T} c_t h_t(x_i)$$

which is between –1 and +1. The quantity c is simply the normalized version of weight b of the hypothesis h, i.e. c=b/|b|. The larger the margin the better generalization performance the combined classifier has. Taking a close look at mechanism of AdaBoost, we will see that what AdaBoost does is actually an asymptotical minimization to a function of the margin:

$$g(\mathbf{b}) = \sum_i \exp\left\{-\frac{|\mathbf{b}|}{2} mg(z_i, \mathbf{c})\right\}$$

where $|\mathbf{b}| = \sum_t b_t$. Interestingly, we have

$$w_t(z_i) = \frac{\partial g(b_{t-1}) / \partial mg(z_i, b_{t-1})}{\sum_{j=1}^{N} \partial g(b_{t-1}) / \partial mg(z_j, b_{t-1})}$$

which is a gradient of $g(b_{t-1})$ with respect to margins. This $w_t(z_i)$ will give a hypothesis h_t which is an approximation to the optimal hypothesis h_t^* that would be obtained by minimizing g(**b**) directly. Therefore, AdaBoost is essentially an approximate gradient descent method which minimizes g(**b**) asymptotically. As g(**b**) is minimized, the minimum margin of the patterns, ρ, is maximized, i.e. we have reached the optimal hyperplane. This is exactly same as in the SVMs. At that moment

$$mg(z_i, \mathbf{c}) \geq \rho \quad \text{for all } i = 1,...,m.$$

In order to avoid overfitting and to get a good generalization performance, we must put penalty on those high weights for the *difficult* training patterns which lead to overfitting, i.e. we need to do weight decay operation. We introduce a slack variable ζ_i^t:

$$\zeta_i^t = (\sum_{r=1}^{t} c_r w_r(z_i))^2$$

where the inner sum is the cumulative weight of the pattern in the previous iterations. This ζ_i^t gives *difficult* patterns big weights which are far from the average. We want

$$mg(z_i, \mathbf{c}) \geq \rho - C\zeta_i^t \quad \zeta_i^t > 0$$

so that some classification errors would be allowed. Then

$$mg(z_i, \mathbf{c}) + C\zeta_i^t$$

is called *soft margin.* It reflects a tradeoff between the margin and the importance of a pattern in training process. Consequently, we derive a new error function with the soft margin:

$$g_{reg}(c_t, |\mathbf{b}_t|) = \sum_i \exp\left\{-\frac{|\mathbf{b}_t|}{2}(mg(z_i, \mathbf{c}_t) + C\zeta_i^t)\right\}$$

and compute the derivation of g_{reg} subject to margin $mg(z_i, b_{t-1})$, we get the weight

$$w_t(z_i) = \frac{\partial g_{reg}(b_{t-1})}{\partial\left(mg(z_i, c_{t-1}) + C\zeta_i^{t-1}\right)}$$
$$= \frac{\exp\left\{-|b_{t-1}|(mg(z_i, c_{t-1}) + C\zeta_i^{t-1})/2\right\}}{\sum_{j=1}^{m}\exp\left\{-|b_{t-1}|(mg(z_j, c_{t-1}) + C\zeta_i^{t-1})/2\right\}}$$

4.3.1 Regularized AdaBoost

Now we give out the regularized AdaBoost algorithm.

Given: m examples $Z = \{(x_1, y_1),...,(x_m, y_m)\}$

Initialize: $w_1(z_i) = 1/m$ for all $i = 1...\mathrm{m}$

For $t = 1, \ldots, \mathrm{T}$:

- Train classifier on weighted sample set $\{Z, w_t\}$ and obtain hypothesis

$h_t : x \mapsto [-1,+1]$

- Find the weight $w_t(z_i)$ of the hypothesis:

$$b_t = \arg\min_{b_t \geq 0} \sum_{i=0}^{m} \exp\{-\frac{1}{2}[\rho(z_i, \mathbf{b}^t) + C\,|\,\mathbf{b}^t\,|\,\zeta_i^t]\}$$

where $\rho(z_i, \mathbf{b}^t) \equiv \mathrm{y_i}\sum_{\mathrm{r}=1}^{\mathrm{t}} \mathrm{c_r} h_r(x_i)$

abort if

$$b_t = 0 \text{ or } b_t \geq \Gamma,$$

where Γ is a large constant

- Update:

$$w_{t+1}(z_i) = \frac{w_t(z_i)}{Z_t} \exp\{-\frac{1}{2}[\rho(z_i, \mathbf{b}^t) + C\,|\mathbf{b}^t|\,\zeta_i^t]\}$$

where Z_t is a normalization constant, such that $\sum_{i=1}^{m} w_{t+1}(z_i) = 1$.

Output final hypothesis:

$$H(x) = sign(\sum_{t=1}^{T} b_t h_t(x)).$$

Figure 4.5: Pseudocode of regularized AdaBoost

C is regularization constant. While C = 0, this algorithm is equivalent to standard AdaBoost.

4.4 Performance Evaluation

In order to evaluate the performance of the ensemble of SVMs, we made a comparison between SVM and standard AdaBoost based on SVM. In the boosting algorithm, the kernel of the base SVM was the same as that of the single SVM, only with this setting that we could compare the difference of performance between the single SVM and the committee of SVMs.

4.4.1 Datasets

The reason for that people usually use only *weak* learner as the base of ensemble is that when the base learner has high accuracy on the training set, the benefit of ensemble will be small. For example, if a single SVM has 95% accuracy on a given dataset, then even if the ensemble of SVMs could reach the accuracy of 96% or 97%, it wouldn't make much significant sense, because the single SVM has almost reached the *'ceiling'* of the performance, in other word, it is already good *enough*. So, in order to test the efficiency of the ensemble of SVMs, we must choose those datasets on which the single SVM

performs relatively *poorly*, e.g. a 70% accuracy. In those cases, the developing space for the ensemble of SVMs is still rather large.

For simplicity, we did not perform experiment on the standard UCI datasets. We only test the algorithm on two-dimensional two-class datasets, i.e. planar point classification problem. Each instance is represented by two attributes (X and Y coordinates), and belongs to one of two classes (represented by +1 and –1, respectively), as illustrated in the following figure:

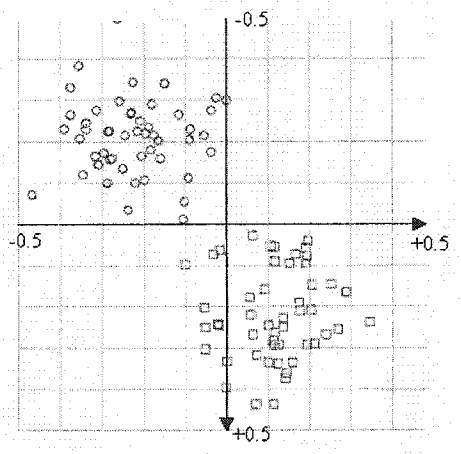


Figure 4.6: Data illustration

Note: in our experiments, the *'circle'* points have class label '-1', and the *'square'* points have class label '+1'.

In fact, looking at the nature of the UCI data, we can see that they are actually the same as our artificial data, because in UCI case, an instance can also be represented as a point in a high-dimensional space. So the results we obtain on current datasets keep the correctness and usefulness on more complex datasets.

We used five above-like artificial datasets. We named them 4_corners, checkers_9, T_noise, the_letter_S and xor_200, respectively. The distributions of their data points

reflect different degree of difficulty. The geometric appearance of the distribution and statistical characteristics are listed out in the following:

1. 4_Corners

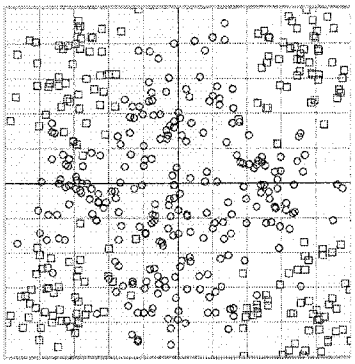


Figure 4.7: 4 Corners

Characteristics: 250 circle points clustered in the central part, 200 square points scattered at the four corners. There are 450 instances in all.

2. Checkers_9

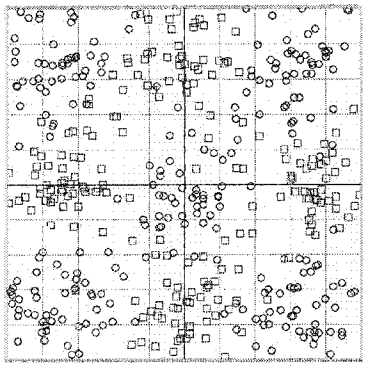


Figure 4.8: Checkers 9

Characteristics: 250 circle points, 200 square points. Each cluster of points of one class is tessellated by a cluster of points of another class. No clusters of points of same class are neighboured. There are 450 instances in all.

3. T_noise

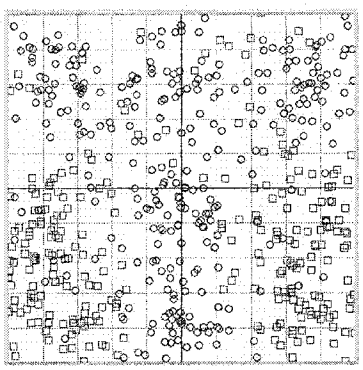


Figure 4.9: T noise

Characteristics: 300 circle points, 250 square points. There are 550 instances in all. Circle points distribute in a 'T' shape, and some square points are mixed in the 'T' area.

4. The_letter_S

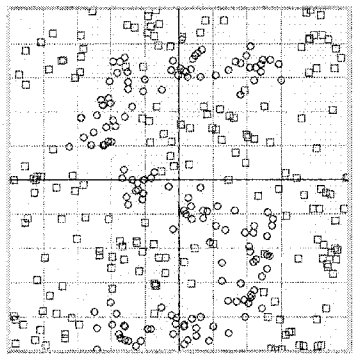


Figure 4.10: The letter S

Characteristics: 150 circle points, 180 square points. There are 330 instances in all. Circle points distribute in an 'S' shape.

5. Xor_200

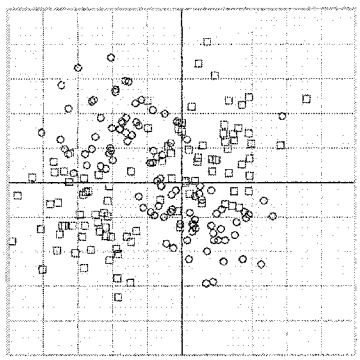


Figure 4.11: Xor 200

Characteristics: 100 circle points, 100 square points. There are 200 instances in all. This is obviously an XOR problem.

4.4.2 Implementation

The algorithm that uses the standard AdaBoost method to combine multiple SVMs has been implemented. The regularized AdaBoost algorithm has not been implemented. The implementation and experiments were done on the platform of Matlab 6.0.

Training set: We did not use the k-fold cross validation. This is because of the characteristics of our datasets. For example, taking a look at the distribution figure of Xor_200 dataset, the data points are distributed in four separated areas. If we do k-fold cross validation, it is very possible that the data points in the test fold all come from the same area, while few point of same class falls in the training set, such that the test error might go up to near 100%. In order to avoid this problem, we used sampling method which randomly draws points from the original dataset. This can guarantee that all data points have the same chance to appear in both training set and test set. 2/3 of each dataset are drawn to form the training set, the remaining forms the test set. When training error is equal to zero or larger than ½, training will have to be stopped.

Test set: After the training set is taken out, the remaining 1/3 of the whole dataset is used as test set.

Kernel function of SVM: we employed SVM of polynomial kernel as the base classifier in our experiments. Three different degree of polynomial kernel function were used: d = 1 (i.e. linear SVM), d = 2 and d = 4. However, the complexity of the problem might be out of our expectation, the distribution of those data points in the high-dimension vector space might be very complex, in order to improve recognition accuracy, employing nonlinear kernel functions, such as the Gaussian Radial Basis Function, might be necessary in future works. For constant C, we tried different values from low to high, the results given out below corresponds to C = 1000 and C = 10. These two C values represent the case of low tolerance of error and the case of high tolerance of error, respectively.

4.4.3 Results

We used the SVM with polynomial kernel as the base of AdaBoost, which is the same as the single SVM classifier. The comparison between two methods: single SVM, standard AdaBoost basing on the same SVM. The generalization error (in %) and iteration numbers are listed out in the following tables. The iteration number reflects how many single SVM are combined into the formation of the ensemble of SVMs.

d = 1 C = 1000	SVM	Ensemble of SVMs	
	Test error	Test error	Iteration no.
4_corners	42	36	2
Checkers_9	46.7	46.7	1
T_noise	31.1	31.1	1
The_letter_S	41	41	1
Xor_200	44.8	44.8	5

d = 2 C = 1000	SVM	Ensemble of SVMs	
	Test error	Test error	Iteration no.
4_corners	14.7	14.7	8
Checkers_9	35.3	32	16
T_noise	13.7	13.7	8
The_letter_S	26.4	23.6	21
Xor_200	10.4	10.4	10

d = 4 C = 1000	SVM	Ensemble of SVMs	
	Test error	Test error	Iteration no.
4_corners	6.7	6.7	17
Checkers_9	10	9.3	28
T_noise	12.6	12.6	16
The_letter_S	13.6	5.4	13
Xor_200	14.9	14.9	14

d = 1 C = 10	SVM	Ensemble of SVMs	
	Test error	Test error	Iteration no.
4_corners	40	30.7	18
Checkers_9	40.7	38.7	3
T_noise	23	23	3
The_letter_S	41.8	38.2	17
Xor_200	37.3	26.9	12

Table 4.1: Performance of ensemble of SVMs on artificial data

4.4.4 Results Analysis

From the above results, we can see an interesting feature: the performance of the ensemble of SVMs is ***at least*** equally good as that of the single SVM. That is to say, it

will not be worse than SVM. The feature is probably from the following fact: during the procedure of constructing multiple SVM, the SVM solution obtained in the first iteration is in fact the same as the single SVM (because the weight of all data points is 1 in the first iteration), and the weight of the first SVM is the largest in all of SVMs (this is because the first SVM has the lowest error rate). Since the final hypothesis is a weighted combination of these SVMs, the one with the largest weight will obviously give most affect to the decision of final hypothesis. In fact, some other researchers had the same observation: the performance of a parallel mixture of SVMs is at least as good as one SVM (Collobert et al, 2002).

Another feature is that we did not get large amount of iteration numbers, unlike in other combining methods, such as the AdaBoost using decision stump as the base, usually those methods can combine up to thousands of base learners. The reason for this feature is that since in the experiments we used hard margin SVM algorithm, not the soft margin one, it treats the error so strictly that the error rate increases very fast with the increase of iteration numbers. When the error rate overcomes ½, the ensemble procedure had to stop immediately. So that is why we did not get many iteration numbers. This also gives another explanation from lateral why the ensemble performs at least equally well as a single SVM, because the weighted decision of the first-iteration SVM is much bigger than the sum of all other weighted decisions, since we have only a few iterations.

From the 4th table we can see that while C takes a small value, the results of the ensemble method are significantly better than that while C take a big value. As we know, C is the regularization parameter in SVM; smaller C corresponds to more tolerant to the error that SVM makes. So it is not surprised that we can obtain better results on noisy data when setting C small.

4.4.5 Discussion

Our preliminary experiments showed that the ensemble of SVMs is promising. It might have a big potential in classification problem research. Although we did not implemented the regularized AdaBoost yet, we could foresee that it could perform better than our current ensemble method used in this experiment, because the essence of this method is to achieve a soft margin (through regularization term and slack variables) in contrast to the hard margin classification. The soft-margin approach allows controlling how much we trust the data, so we are permitted to ignore noisy patterns (e.g. outliers) that would otherwise have spoiled our classification. Thus, it can avoid overfitting and get better generalization performance. The benefits of ensemble methods are quite attractive, but there is no free lunch! The disadvantages of it are also quite obvious. There are mainly two disadvantages about the ensemble of SVMs comparing to the single SVM:

- Much larger memory is required
- Much longer computation time is needed

These are due to the fact that in the ensemble method we have to construct N SVM solutions on N different weighted dataset. This limitation may sometimes make it impractical on large-scale dataset.

CONCLUSION

This project is mainly an exploratory and preliminary study of the scoliosis estimation problem by making using of the surface deformity information. We proposed a new method to the old problem of estimating the severity of scoliotic spinal deformity from the external trunk surface. Our starting point was to avoid the two limitations in the GA-ANN method, so we did not make any feature extraction and utilized the control points of the surface model of the trunk as the representation, and we replaced ANN by SVM. But unfortunately our results were not as good as Jaremko's on the Calgary data set. The reasons were various. From the GA-SVM experiments, we proved that SVM performed as well as ANN. So we think the main problem was from using the control points as the representation of the surface deformities. As we discussed (see section 3.7), control point is not a kind of invariant and steady representation of the surface deformities. And experimental results showed that control points did not capture the surface deformities as accurately as the features used by Jaremko. We took into account the control points of the whole trunk. In fact this is not necessary. Many control points were redundant and should not be kept as a part of the data, but the practical difficulty was that we could not tell which control point was important and which was not. A possible refinement is to use only the control points of the part of trunk between arm-pit and waist. GA-ANN method firstly also suffered from the problem of taking in account the features of the whole trunk, later Jaremko focused only on the features extracted from the part between arm-pit and waist, and then he obtained the optimal results. Another possible refinement is to use only the control points of the back of the trunk. This is because most significant deformities appeared on the back instead of the front, and the breast part of female patients can lead to very different control points from male patients.

The other main reason was the small dataset and the data set was full of all kind of noises (see section 3.2). The feature extraction method used in Jaremko's GA-ANN approach can less suffer from these noises, but our method cannot, because we made full use of all

the data points directly and the control points do not capture the surface deformities as accurately as the features. With our approach, the dimensionality of the data set was very high. Although we utilized a type of dimensionality reduction technique, PCA, the problem was still not solved. Some information was lost during the dimension reduction procedure. When more patients' data of better quality will be collected, our method can be expected to perform better. The ideal way is to divide patients into as more subgroups as possible with differing age, gender, curve severity, curve type and bracing status. Thus each subgroup categorizes precisely a sort of curve type of the spine, and there are enough data in each subgroup. But this is only feasible under the a priori condition of having sufficient data. The advantages of our method are mainly as we discussed in (section 1.6.2), i.e., that it is general and applicable in practice; it can avoid the two main limitations in the GA-ANN method (see section 1.6.1); and it is very fast in training and testing compared to GA-ANN method. The main disadvantage is that it predicted the Cobb angle less accurately than the GA-ANN method. SVM classification was very fast in both training and testing, even when the PCA was not applied onto the datasets. But our parameter tuning procedure was quite slow, for instance, on the 61x31 dataset the algorithm took hours of computation time to get the optimal parameter value.

In this project, several techniques have been tried and their usefulness in our problem has been discussed. These experiences can be used as a reference and guide for a future study.

As for the future study, we think the main direction is to seek for another type of steady and invariant representation of the surface deformities other than the control points. For instance, the derivative of the surface model is relatively much steadier than the control points. It is invariant to proportional changes to the surface model. Thus, even though a young patient might become taller or stronger at the second scanning in half a year, the derivatives (of several important regions, for example) of his trunk surface very possibly remains unchanged or only slightly changed if his spine did not progress significantly during this period.

Another important need to be solved is to determine how many control points are sufficient for representing the raw geometrical points. In our experiments, we tried three sets of density: 16x8, 41x11, and 61x31. These three types of density correspond to loose, moderate, and tight fitting, respectively. We found that increasing the density of control points did not improve the classification accuracy. On the contrary, the best and the second best results were found on the loosest density, i.e., the 16x8 datasets. But even 16x8 might not be the optimal number for the density of control points. We need to develop a kind of method which can automatically search the optimal number of control points, like in the parameter tuning procedure, if we still continue in the direction of using control points as the mere representation of surface deformities. But if we go another direction in which we employ other type of steady and invariant representation such as the derivatives of the surface model, then the problem of determining how many control points are sufficient can be solved. What we have to do now is to increase control points until we fit the surface to a user-specified tolerance range, for example, 5% error. Repeat this procedure on each patient and the maximum number of control points is the optimal number for all the patients for that user-specified tolerance.

Instead of the linear PCA method employed in our method, other types of dimension reduction methods are also worth exploring. For instance, nonlinear PCA, Multi-Dimensional Scaling (MDS), Locally Linear Embedding (LLE) etc. Low-dimensional representation of data is also beneficial.

Instead of the SVM kernels utilized in our experiments (including linear, polynomial, RBF, and ERBF), other kernels, such as sigmoid, Fourier series, B splines, additive kernels, tensor product kernels etc., are also worth exploring, although the ones we utilized are the most popular ones. A method which can automatically find out the optimal kernel basing on our dataset is demanded too.

In constructing the ensemble of SVMs, in addition to these general-purpose ensemble methods, such as boosting and bagging, there are several other algorithm-specific methods for generating ensembles reported.

- Rosen (1996) trains several neural networks simultaneously and forces the networks to be diverse by adding a correlation penalty to the error function that back-propagation minimizes. He reports substantial improvements in three synthetic tasks.
- Opitz and Shavlik (1996) take a similar approach, but they employ a kind of genetic algorithm to search for a good population of neural network classifiers. In a comparison with bagging, they found that their method gave excellent results in four real-world domains.
- Bayesian model averaging (Madigan et al. 1996)
- Averaging over models output by a randomized learning algorithm
- LocBoost
- etc.

It would be worth exploring the applicability and efficiency of these methods in combining SVMs.

BIBLIOGRAPHY

Akay YM, Akay M., Welkowitz W, and Kostis JB, (1994). *Noninvasive detection of coronary artery disease using wavelet-based fuzzy neural networks.* IEEE Engineering in Medicine and Biology, 761-764.

Aubin C.E., Descrimes J.L., Dansereau J., Skalli W., Lavaste F., Labelle H. (1995). *Geometrical modeling of the spine and thorax for biomechanical analysis of scoliotic deformities using finite element method (in French).* Ann. Chir. 49(8), 749-761.

Batouche M and Benlamri R, (1994). *A computer vision system for diagnosing scoliosis.* Systems, Man, and Cybernetics, 1994. Humans, Information, and Technology, 1994 IEEE International Conference on, Vol. 3, 1994, page(s) 2623 -2628 Vol. 3

Bourlas Ph, Sgouros N, Papakonstantinou G, and Tsanakas P, (1996). *Towards a knowledge acquisition and management system for ECG diagnosis.* In Proceedings of 13th International Congress Medical Informatics Europe-MIE96, Copenhagen.

Bratko I, Mozetic I, and Lavrac N, (1989). *KARDIO: A study in deep and qualitative knowledge for expert systems.* Cambridge, Massachusetts: MIT Press.

Breiman L, (1996). *Bias, variance, and arcing classifiers.* Technical Report 460, Statistics Department, University of California.

Bunnell WP, (1984). *An objective criterion for Scoliosis Screening.* J Bone and Joint Surg 66A: 1381-1387.

Burbidge R, Trotter M, Buxton B, and Holden S, (2001). *Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis*. Elsevier Science, April.

Cho SB, and Kim JH, (1997). *Combining Multiple Neural Networks by Fuzzy Integral for Robust Classification.* IEEE Trans, Systems, Man, and Cybernetics, vol. 25, no. 2, pp. 380-384, 1995. Proc. 14th Int'l Conf. Machine Learning, 1997.

Chris Ding, and Inna Dubchak, (2001). *Multi-class Protein Fold Recognition Using Support Vector Machines and Neural Networks.* Bioinformatics, April 2001, Vol 17, No 4, pp.349-358.

Christopher J.C. Burges, (1998). *A tutorial on support vector machines for pattern recognition.* Knowledge Discovery and Data Mining, 2(2), 1998.

Closkey RF, and Schultz AB, (1993). *Rib cage deformities in scoliosis: spine morphology, rib cage stiffness, and tomography imaging.* Journal of Orthopedic Research, 11, 730-737.

Collobert R, Bengio S, and Bengio Y, (2002). *A parallel mixture of SVMs for very large scale problems*. Neural Computation, vol. 14, no. 5.

Cook LT, De Smet AA, Tarlton MA, and Fritz SL, (1981). *Assessment of scoliosis using three-dimensional analysis.* IEEE Transactions on Biomedical Engineering, BME-28, start page 366.

Coomans, D., Broeckaert, M. Jonckheer M. and Massart D.L., (1983). "*Comparison of Multivariate Discriminant Techniques for Clinical Data - Application to the Thyroid Functional State*", Meth. Inform. Med. 22 (1983) pp. 93-101.

Coomans, D. and I. Broeckaert, (1986). "*Potential Pattern Recognition in Cemical and Medical Decision Making*", Research Studies Press, Letchworth, England.

Coppini G, Poli R, and Valli G, (1995). *Recovery of the 3-D shape of the left ventricle from echocardiographic images.* IEEE Transactions on Medical Imaging, 14, 301-317.

Dansereau J, Stokes IAF, (1988). *Radiographic reconstruction of 3-D human rib cage.* J Biochech 21: 893-901.

Dansereau et al. (1990). *Three-dimensional reconstruction of the spine and rib cage from stereoradiographic and imaging techniques.* Proc. of the CSME Forum, Toronto, Canada; p.61-64.

Dasarathy, B.V. (1980) "*Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments*". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71.

Dawant BM, Ozkan M, Sprenkels H., Aramata H., Kawamura K, and Margolin RA, (1990). *A neural network approach to magnetic resonance imaging tissue characterization.* Communication, Control, and Signal Processing, Arikan E., (ed.), 2, 1803-1809, Bilkent University, Ankara, Turkey, Elsevier, Amsterdam.

Dawson EG, Kropf MA, Purcell G, Kabo JM, Kanim LE, and Burt C, (1993). *Optoelectronic evaluation of trunk deformity in scoliosis.* Spine, 18(3): 326-331.

Delaney PM, Papworth GD, and King RG, (1998). *Fibre optic confocal imaging (FOCI) for in vivo subsurface microscopy of the colon.* In Methods in disease: Investigating the Gastrointestinal Tract, Preedy VR and Watson RR (eds.), Greenwich Medical Media, London.

Denton TE, et al, (1992). *Spine.* 17(5): 509-512.

Dietterich TG, (2000). *An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization.* In Machine Learning, 40(2):139-158.

Drerup B and Hierholzer E, (1996). *Assesment of scoliotic deformity from back shape asymmetry using an improved mathematical model.* Clin Biomech, 11(7): 376-383.

Duan K, Keerthi SS and POO AN, (2001). *Evaluation of simple performance measures for tuning SVM hyperparameters.* A technical report. Department of Mechanical Engineering, National University of Singapore, 2001.

Edward Jackson J, (1991). *A user's guide to principal components.* John Wiley & Sons, Inc.

Fernandez R, (1999). *Predicting time series with a local support vector regression machine.* ACAI99.

Gates, G.W. (1972) "*The Reduced Nearest Neighbor Rule*". IEEE Transactions on Information Theory, May 1972, 431-433.

Gindi GR, Darken CJ, O' Brien KM, Sterz ML, and Deckelbaum LI, (1991). *Neural network and conventional classifiers for fluorescence-guided laser angioplasty.* IEEE Transactions on Biomedical Engineering, 38, 3, 246-252.

Goldberg CJ. Kaliszer M. Moore DP. Fogarty EE. Dowling FE. (2001a). *Surface topography, Cobb angles, and cosmetic change in scoliosis.* Spine, 26(4): E55-63.

Gomes AS. Serra LA. Lage AS. Gomes A (1995). *Automated 360° profilometry of human trunk for spinal deformity analysis.* Three-dimensional analysis of spinal deformities. M. D'Amico (Ed.). Amsterdam, NL, IOS Press: 423-429.

Gunn SR, Brownand M, and Bossley KM, (1997). *Network performance assessment for neurofuzzy data modelling.* Lecture notes in computer science, 1280: 313-323.

Gunnar Ratsch, Bernhard Scholkopf, Sebastian Mika, and Klaus Robert Muller, (2000). *SVM and Boosting: One Class.* GMD Report No. 119, November 2000, 36 pages.

Gunnar Ratsch, Takashi Onoda, and Klaus Robert Muller, (1998). *An improvement of AdaBoost to avoid overfitting.* In Proc. of the Int. Conf. on Neural Information Processing (ICONIP), pp. 506-509 Kitakyushu, Japan.

Gunnar Ratsch, Takashi Onoda, and Klaus Robert Muller, (2000). *Soft margin for AdaBoost.* Machine Learning: 1-35 (2000). 2000 Kluwer Academic Publishers, Boston, Manufactured in Netherlands.

Guo Z, Durant LG, Lee HC, Allard L, Grenier MC, and Stein PD, (1994). *Artificial neural networks in computer-assisted classification of heart sounds in patients with porcine bioprosthetic valves.* Medical, Biological Engineering and Computing, 32, 311-316.

Haasbeek J, (1997). *Postgraduate Medicine.* 101(6): 207-16.

Hanka R., Harte TP, Dixon AK, Lomas DJ, and Britton PD, (1996). *Neural networks in the interpretation of contrast-enhanced magnetic resonance images of the breast.* In Proceedings of Healthcare Computing, Harrogate, UK, 275-283.

Hansen LK and Salamon P, (1990). *Neural Network Ensembles.* IEEE Trans, Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993-1,001, Oct. 1990.

Hashem and B. Schmeiser, (1995). *Improving Model Accuracy Using Optimal Linear Combinations of Trained Neural Networks.* IEEE Trans, Neural Networks, vol. 6, no. 3, pp. 792-794, 1995.

Hau D and Coiera E, (1997). *Learning qualitative models of dynamic systems.* Machine Learning, 26, 177-211.

Herzenberg JE, Waanders NA, Closkey RF, Schultz AB, and Hensinger RN, (1990). *Cobb angle versus spinous process angle in adolescent idiopathic scoliosis. The relationship of the anterior and posterior deformities.* Spine, 15(9): 874-879.

Hoffman DA, Lonstein JE, Morin MM, Visscher W, Harris BSd, Boice JD, JR, (1989). *Breast cancer in women with scoliosis exposed to multiple diagnostic X-rays.* J Natl Cancer Inst, 81(17): 1307-1312.

Hyoung Seop Kim, Ishikawa S, Ohtsuka Y, Shimizu H, Shinomiya T, Viergever MA, (2001). *Automatic scoliosis detection based on local centroids evaluation on moire topographic images of human backs.* Medical Imaging, IEEE Transactions on, Vol. 20 Issue 12, Dec. 2001 Page(s) 1314 –1320.

Ifeachor EC and Rosen KG (eds.), (1994). *Proceedings of the International Conference on Neural Networks and Expert Systems in Medicine and Healthcare.* Plymouth, UK.

Innocent PR, Barnes M, and John R, (1997). *Application of the fuzzy ART/MAP and MinMax/MAP neural network models to radiographic image classification.* Artificial Intelligence in Medicine, 11, 241-263.

Ishida A. Mori Y. Kishimoto H. Nakazima T. Tsubakimoto H. (1987). *Body shape measurement for scoliosis evaluation.* Med Biol Eng Comput, 25(5): 583-585.

Ishida A. Suzuki S. Imai S. Mori Y. (1982). *Scoliosis evaluation utilising truncal cross-sections.* Med Biol Eng Comput, 20(2): 181-186.

Jaremko JL, (2001). *Estimation of scoliosis severity from the torso surface by neural networks.* PhD thesis. University of Calgary.

Jaremko JL, (2002). *Genetic algorithm-neural network estimation of cobb angle from torso asymmetry in scoliosis.* Transactions of the ASME, journal of biomechanical engineering Vol. 124, page 496 – 503, October 2002.

Jefferson MF, Pendleton N, Lucas SB, and Horan MA, (1997). *Comparision of a genetic algorithm neural network with logistic regression for predicting outcome after surgery for patients with nonsmall cell lung carcinoma.* Cancer, 79(7): 1338-1342.

Joachims T, (1999). *Making large-scale support vector machine learning practical.* Advances in kernel methods: support vector learning. 1999.

John C. Platt, (1998). *Fast Training of Support Vector Machines using Sequential Minimal Optimization.*

Jolliffe IT, (1986). *Principal component analysis.* Springer-Verlag.

Kalifa G, et al, (1998). *Pediatric Radiology.* 28: 557-561.

Karkanis S, Magoulas GD, Grigoriadou M, and Schurr M, (1999). *Detecting abnormalities in colonoscopic images by textural description and neural networks.* In

Proceedings of Workshop on Machine Learning in Medical Applications, Advance Course in Artificial Intelligence-ACAI99, Chania, Greece, 59-62.

Keerthi SS, Lin CJ, (2002). *Asymptotic behaviors of support vector machines with gaussian kernel.* To appear in Neural Computation. The MIT Press.

Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK, (1999). *Improvements to Platt's SMO Algorithm for SVM Classifier Design.*

Keerthi SS, (2001). *Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms.* Control division technical report. National University of Singapore, 2001.

Kennedy LR, Harrison RF, Burton AM, Fraser HS, Hamer WG, MacArthur D, McAllum R, and Steedman DJ, (1997). *An artificial neural network system for diagnosis of acute myocardial infarction (AMI) in the accident and emergency department: evaluation and comparison with serum myoglobin measurements.* Computer Methods and Programs in Biomedicine, 52, 93-103.

King HA, Moe JH, Bradford DS, Winter RB, (1983). *The selection of fusion levels in thoracic idiopathic scoliosis.* J. Bone Joint Surg., 65-A (9): 1302-1313.

Kralj K and Kuka M, (1998). *Using machine learning to analyze attributes in the diagnosis of coronary artery disease.* In Proceedings of Intelligent Data Analysis in Medicine and Pharmacology-IDAMAP98, Brighton, UK.

Krogh A and Vedelsby J, (1995). *Neural Network Ensembles, Cross Validation, and Active Learning.* Advances in Neural Information Processing Systems 7, G. Tesauro, Touretzky DS, and Leen TK, eds. Cambridge, Mass.: MIT Press.

Kwok JT, (1998). *Support vector mixture for classification and regression problems.* In proceedings of the International Conference on Pattern Recognition (ICPR), page 255-258, Brisbane, Queensland, Australia.

Labelle et al. (1995). *Variability of geometric measurements from 3-D reconstructions of scoliotic spines and rib cages.* Eur Spine J; 4, p. 88-94.

Labelle H., Dansereau J., Bellefleur C., Poitras B. (1996). *Three-dimensional effect of the Boston brace on the thoracic spine and the rib cage.* Spine 21-1, 59-64.

Laurent-Gengoux P and Mekhilef M, (1993). *Optimization of a NURBS representation.* CAD, Vol. 25, No. 11, pp. 699-710.

Lavrac N, (1998). *Data mining in medicine: Selected techniques and applications.* In Proceedings of Intelligent Data Analysis in Medicine and Pharmacology-IDAMAP98, Brighton, UK.

Leo Breiman, (1996). *Bagging predictors.*

Letts M, Quanbury A, Gouw G, Kolsun W, and Letts E, (1988). *Computerized ultrasonic digitization in the measurement of spinal curvature.* Spine, 13(10): 1106-1110.

Levy AR, Goldberg MS, Mayo NE, Hanley JA, and Poitras B, (1996). *Reducing the lifetime risk of cancer from spinal radiographs among people with adolescent idiopathic scoliosis.* Spine, 21(13), 1540-1548.

Lim CP, Harrison RF, and Kennedy RL, (1997). *Application of autonomous neural network systems to medical pattern classification tasks.* Artificial Intelligence in Medicine, 11, 215-239.

Liu X, Thometz J, Lyon R, and Klein J, (2001). *Functional classification of patients with idiopathic scoliosis assessed by the Quantec system.* Spine, 26(11): 1274-1279.

Lo SCB, Lou SLA, Lin JS, Freedman MT, Chien MV, and Mun SK, (1995). *Artificial convolution neural network techniques and applications for lung nodule detection.* IEEE Transactions on Medical Imaging, 14, 711-718.

Lovell FW, Rothstein JM, and Personius WJ, (1989). *Reliability of Clinical Measurements of Lumbar Lordosis taken with a flexible rule.* Physical Therapy 69 (2): 342-345.

Man Leung Wong, Wai Lam, Kwong Sak Leung, Po Shun Ngan, Jack C.Y. Cheng, (2000). *Discovering knowledge from medical databases using evolutionary algorithms.* IEEE Engineering in Medicine and Biology, July/August 2000.

Marshall SJ, Rixon RC, Whiteford DN, and Cumming JT, (1998). *The OrthoForm 3-Dimensional Clinical Facial Imaging System.* Proc. Int. Fed. Hosp. Eng. Congress 15, 83-87.

Marzan G.T. (1976). *Rational Design for Close-Range Photogrammetry.* Thèse de Ph.D., Département de Génie Civil, Université de l'Illinois, Urbana-Champaign, USA

Micheli-Tzanakou E, Yi C, Kostis WJ, Shindler DM, and Kostis JB, (1993). *Myocardial infarction: Diagnosis and vital status prediction using neural networks.* IEEE Computers in Cardiology, 229-232.

Miller AS, Blott BH, and Hames TK, (1992). *Review of neural network applications in medical imaging and signal processing.* Medical and Biological Engineering and Computing, 30, 449-464.

Moe JH, (1958). *A critical analysis of methods of fusion for scoliosis: An evaluation of two hundred and sixty-six patients.* J Bone Joint Surg. 1958; 40-A: 529-554.

Moreland MS, Pope MH, Wilder DG, Stokes I, and Frymoyer JW, (1981). *Moire fringe topography of the human body.* Medical Instrumentation, Vol. 15, start page 129.

Muller K, Smola A, Ratsch G, Scholkopf B, Kohlmorgen J, and Vapnik V, (1997). *Predicting time series with support vector machines.* In proceedings of the International Conference on Artificial Neural Networks.

Nash CL, Gregg EC, Brown RH, and Pillac K, (1979). *Risks of Exposure to X-Rays in Patients Underging Long Term Treatment for Scoliosis.* J Bone and Joint Surg 61A, 371.

National Scoliosis Foundation. http://www.scoliosis.org/

Nekovei R and Sun Y, (1995). *Back-propagation network and its configuration for blood vessel detection in angiograms.* IEEE Transactions on Neural Networks, 6, 1, 64-72.

Osuna E, Freund R, and Girosi F, (1997). *Support Vector Machines: Training and Applications.* A.I. Memo No. 1602, Artificial Intelligence Laboratory, MIT.

Oxford Metrics, (1987). *ISIS Operating Manual Version 4.0, 1.1.*

Pattichis C, Schizas C, and Middleton L, (1995). *Neural network models in EMG diagnosis.* IEEE Transactions on Biomedical Engineering, 42, 5, 486-496.

Phee SJ, Ng WS, Chen IM, Seow-Choen F, and Davies BL, (1998). *Automation of colonoscopy part II: visual-control aspects.* IEEE Engineering in Medicine and Biology, May/June, 81-88.

Piegl LA and Tiller W, (1994). *The NURBS book.*

Piegl LA and Tiller W, (2000). *Surface approximation to scanned data.* The visual computer (2000) 16:386-395.

Poncet P, Delorme S, Ronsky JL, Dansereau J, Harder J, Clynch G, Dewar RD, Labelle H, Gu PH, Zernicke RF, (2000a). *Reconstruction of laser-scanned 3D torso topography and stereo-radiographical spine and rib-cage geometry in scoliosis.* Comp Meth Biol Biomed Eng, 4(1): 59-75.

Pope MH, Stokes IAF, Moreland M, (1984). *The biomechanics of scoliosis.* CRC Crit Rev Biomed Eng, 11(3): 157-188.

Prentza A and Wesseling KH, (1995). *Catheter-manometer system damped blood pressures detected by neural nets.* Medical and Biological Engineering and Computing, 33, 589-595.

Rayburn DB, Klimasauskas CC, Januszkievicz AJ, Lee JM, Ripple GR, and Snapper JR, (1990). *The use of back propagation neural networks to identify mediator-specific cardiovascular waveforms.* In Proceedings of the International Joint Conference on Neural Networks, 2, 105-110.

Reategui EB, Campbell JA, and Leao BF, (1996). *Combining a neural network with case-based reasoning in a diagnostic system.* Artificial Intelligence in Medicine, 9, 5-27.

Richardson ML, (2001). *Approaches to Differential Diagnosis in Musculoskeletal Imaging.* University of Washington Department of Radiology

Roger RE, Stokes IAF, Harris JD, Frymoyer JW, and Ruiz C, (1979). *Monitoring adolescent idiopathic scoliosis with Moire fringe photography.* Engineering in Medicine, Vol. 8, start page 119.

Rogova G, (1994). *Combining the Results of Several Neural Network Classifiers*, Neural Networks, vol. 7, no. 5, pp. 777-781.

Sakka SA, and Mehta MH, (1997). *Journal of Bone and Joint Surgery.* 79-B (Supp 1): 112.

Schapire RE, Freund Y, Bartlett P, and Lee WS, (1998). *Boosting the margin: A new explanation for the effectiveness of voting methods.* In The Annals of Statistics, 26(5): 1651-1686.

Scholkopf B, Sung K, Burges C, Girosi F, Niyogi P, Poggio T, and Vapnik V, (1996). *Comparing support vector machines with Gaussian kernels to radial basis function classifiers.* A.I. Memo 1599, MIT, Dec 1996.

Sciandra J. De Mauroy JC. Rolet G. Kohler R. Creach JP (1995). *Accurate and fast non-contact 3-D acquisition of the whole trunk.* Three-dimensional analysis of spinal deformities. M. D'Amico (Ed.) Amsterdam, NL, IOS Press: 81-85.

Scoliosis Research Society. http://www.srs.org/homepage.htm

Scutt ND, Dangerfild PH, and Dorgan JC, (1996). *The relationship between surface and radiological deformity in adolescent idiopathic scoliosisL effect of change in body position.* Eur Spine J, 5(2): 85-90.

Stokes IAF, Bigalow LC, and Moreland MS, (1987). *Three-dimensional spinal curvature in idiopathic scoliosis.* J Orthop Res 5: 102-113, 1987.

Stokes IAF, Cobb LC, Pope MH, and Moreland MS. *Back surface shape relationship to spinal deformity.*

Stokes IAF, Dansereau J, and Moreland MS, (1989). *Rib cage asymmetry in idiopathic scoliosis.* Journal of Orthopedic Research, 7, 599-606.

Stokes IAF and Moreland MS, (1989). *Concordance of back surface asymmetry and spine shape in idiopathic scoliosis.* Spine, 14(1): 73-78.

Strausberg J and Person M, (1999). *A process model of diagnostic reasoning in medicine.* International Journal of Medical Informatics, 54, 9-23.

Takasaki H, (1979). *The development and the present status of moire topography.* Optica Acta, Vol. 26, start page 1009.

Tartaro M and Austin JH, (1986). *Moire Topography in Scoliosis. Accuracy of assessing lateral curvature as a function of the region of the curve.* In 'Surface topography and spinal deformity'. Harris JD and Turner-Smith AR eds, pp125-133, Gustav-Fischer Stuttgart, New York.

Tefas A, Kotropoulos C, and Pitas I. (1999). *Enhancing the performance of elastic graph matching for face authentications by using Support Vector Machines.* ACAI99.

Theologis TN. Fairbank JC. Turner-Smith AR. Pantazopoulos T. (1997). *Early detection of progression in adolescent idiopathic scoliosis by measurement of changes in back shape with the Integrated Shape Imaging System scanner.* Spine, 22(11): 1223-1227.

Thometz JG. Lamdan R. Liu XC. Lyon R. (2000). *Relationship between Quantec measurement and Cobb angle in patients with idiopathic scoliosis.* J Pediatr Orthop, 20(4): 512-516.

Thulborne T and Gillespie R, (1976). *The Rib Hump in Idiopathic Scoliosis. Measurement analysis and response to treatment.* J Bone and Joint Surg 58B: 456-462.

Tredwell SJ. Bannon M. (1988). *The use of the ISIS optical scanner in the management of the braced adolescent idiopathic scoliosis patient.* Spine, 13(10): 1104-1105.

Treuillet S, Lucas Y, Crepin G, Peuchot B, Pichaud JC, (2002). *SYDESCO: a laser-video scanner for 3D scoliosis evaluations.* Research into spinal deformities 3. A. Tanguy and B. Peuchot (Eds.), IOS Press.

Turner-Smith AR, Harris JD, Houghton GR, Jefferson RJ, (1988). *A method for analysis of back shape in scoliosis.* J Biomech, 21(6): 497-509.

Turner-Smith AR, (1988). *A Television/Computer Three Dimensional Surface Shape Measurement System.* J Biomechanics, 21, 6, 515-529.

Upadhyay SS. Burwell RG. Webb JK. (1988). *Hump changes on forward flexion of the lumbar spine in patients with idiopathic scoliosis. A study using ISIS and the Scoliometer in two standard positions.* Spine, 13(2): 146-151.

Vandegriend B. Hill D. Raso J. Durdle N. Zhang Z. (1995). *Application of computer graphics for assessment of spinal deformities.* Med Biol Eng Comput, 33(2): 163-166.

Vapnik V, (1995). *The Nature of Statistical Learning Theory.*

Vapnik V, (1998). *Statistical Learning Theory.*

Veropoulos K, Campbell C, and Learmonth G, (1998). *Image processing and neural computing used in the diagnosis of tuberculosis.* Colloquium on Intelligent Methods in Healthcare and Medical Applications, York, UK.

Veropoulos K, Cristianini N, and Campbell C, (1999). *The Application of Support Vector Machines to Medical Decision Support: A Case Study*, ACAI99.

Wang S, Zhu WY, Liang ZP, (2001). *Shape Deformation: SVM Regression and Application to Medical Image Segmentation.* Proceedings of International Conference on Computer Vision (ICCV), Vancouver, Canada,

Weisz I, Jefferson RJ, Turner-Smith AR, Houghton GR, and Harris JD, (1988). *ISIS scanning: a useful assessment technique in the management of scoliosis.* Spine, 13(4): 405-408.

White AA and Panjabi MM, (1990). *Clinical Biomechanics of the Spine.* Philadelphia: Lippincott JB, Co., 3rd ed., ch. 1.

Willner S, (1981). *Spinal Pantograph-A non-invasive technique for describing kyphosis and lordosis in the thoraco-lumbar spine.* Acta Orthop Scand 52: 525-529.

Wojcik AS, Phillips GF, and Mehta MH, (1994). *Recording of the back surface and spinal shape by Quantec image system: a new technique in the scoliosis clinic.* J. Bone Joint Surg 76B, Supp I.

Wong HK, Balasubramaniam P, Rajan U, and Chng SY, (1997). *Direct spinal curvature digitization in scoliosis screening – a comparative study with Moire contourgraphy.* J Spinal Disorder, 10(3): 185-192.

Yeap TH, Johnson F, and Rachniowski M, (1990). *ECG beat classification by a neural network.* In Proceedings of the 12th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 3, 1457-1458, Philadelphia, Pennsylvania, USA.

Yoav Freud and Robert E. Schapire, (1996). Experiments with a new boosting algorithm.

Zhu Y, and Yan H, (1997). *Computerized tumor boundary detection using a Hopfield neural network.* IEEE Transactions on Medical Imaging, 16, 55-67.

Zupan B, Halter JA, and Bohanec M, (1998). *Qualitative model approach to computer assisted reasoning in physiology.* In Proceedings of Intelligent Data Analysis in Medicine and Pharmacology-IDAMAP98, Brighton, UK.